Approximate Cramér-Rao Bounds for Multiple Target Tracking

A Thesis
Presented to
The Academic Faculty

by

William F. Leven

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2006

Approximate Cramér-Rao Bounds for Multiple Target Tracking

Approved by:

Dr. Aaron D. Lanterman, Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. James McClellan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Magnus Egerstedt
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mark Richards
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Nicoleta Serban
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: March 02, 2006

This dissertation is dedicated to my family.

Stephen, Debra, and Julie Leven:

thank you for your love, encouragement, and support.

ACKNOWLEDGEMENTS

I would like to begin by recognizing Prof. Aaron D. Lanterman for being an excellent advisor. He has been a wonderful mentor whose optimism is contagious even in the face of seemingly discouraging research results. His skills in positive motivation are amazing, and while his intellect may be humbling, I never left his office feeling humbled. Finally, I would like to thank Aaron for his kindness and compassion. His concern for the well-being of his students is extraordinary, and for that, I am deeply grateful.

I would also like to thank my committee members, Prof. James McClellan, Prof. Magnus Egerstedt, Prof. Mark Richards, and Prof. Nicoleta Serban for their time and suggestions. Their suggestions have been vital for making this work accessible and useful for a broader audience.

I would also like to thank Fred Daum of Raytheon whose idea of using SME for bound computation started the bounding aspect of this work. The recommendations and financial support of Dr. Dale Blair of the Georgia Tech Research Institute are greatly appreciated. This work was enabled by startup funds from the School of Electrical and Computer Engineering at the Georgia Institute of Technology, the Demetrius T. Paris Junior Professorship, and in part by the U.S. Office of Naval Research grant N00014-02-1-0564.

I would like to thank my group members Martin Tobias, Kerkil Choi, Ryan Palkki, Jason Dixon, and Lisa Ehrman for their friendship and research discussions. I would especially like to thank Martin for his time spent debugging my code, explaining particle filters, proofreading my papers, and for his friendship.

There are many other students at CSIP that I would like to thank. Kevin Chan has been a great friend since my first Georgia Tech visit in the spring of 2002. Maneli Noorkami has been a close friend ever since we struggled through the DSP hardware course together; she deserves special thanks for proofreading the first draft of this thesis. I would like to thank Badri Narayanan for his help and suggestions on all things mathematical, and Rajbabu

Velmurugan and Volkan Cevher for their assistance on target tracking subjects. On the lighter side, many thanks to the French connection: Nicolas Gastaud, Jérôme Vasseur, and David Boivin for the entertainment they have provided for all of us. I have enjoyed many lunch-time conversations with Mina Sartipi, many athletic endeavors with Vince Emanuele, and I thank Sam Li for being CSIP's elder statesman. I would also like to thank the many other CSIP students for their comradeship.

I would like to thank the CSIP staff for taking care of the daily business of the lab, especially Kay Gilstrap and Christy Ellis for their work during the first part of my stay at CSIP and to Carla Zachary for holding it all together when they were promoted to other positions.

In conclusion, I would like to acknowledge my family. Thank you, Jenny, for being my family in Atlanta for most of my time here, and thank you Bill for being an excellent PhD engineer role-model. Thank you, Julie, for your encouragement, faith, and for not pointing out that you left academia and made it to real adulthood (at least) two years before your older brother. Finally, I offer a special thanks to my parents: thank you for being terrific parents, for your encouragement, and for making this possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The main objective of this dissertation is to develop mean-squared error performance predictions for multiple target tracking. Envisioned as an approximate Cramér-Rao lower bound, these performance predictions allow a tracking system designer to quickly and efficiently predict the general performance trends of a tracking system. The symmetric measurement equation (SME) approach to multiple target tracking (MTT) lies at the heart of our method. The SME approach, developed by Kamen *et al.*, offers a unique solution to the data association problem. Rather than deal directly with this problem, the SME approach transforms it into a nonlinear estimation problem. In this way, the SME approach sidesteps "report-to-track" associations.

Developing performance predictions using the SME approach requires work in several areas: (1) extending SME tracking theory, (2) developing nonlinear filters for SME tracking, and (3) understanding techniques for computing Cramér-Rao error bounds in nonlinear filtering. First, on the SME front, we extend SME tracking theory by deriving a new set of SME equations for motion in two dimensions. We also develop the first realistic and efficient method for SME tracking in three dimensions. Second, we apply, for the first time, the unscented Kalman filter (UKF) and the particle filter to SME tracking. Using Taylor series analysis, we show how different SME implementations affect the performance of the EKF and UKF and show how Kalman filtering degrades for the SME approach as the number of targets rises. Third, we explore the Cramér-Rao lower bound (CRLB) and the posterior Cramér-Rao lower bound (PCRB) for computing MTT error predictions using the SME. We show how to compute performance predictions for multiple target tracking using the PCRB, as well as address confusion in the tracking community about the proper interpretation of the PCRB for tracking scenarios.

CHAPTER I

INTRODUCTION

This thesis presents methods of computing approximate Cramér-Rao lower bounds (CRLB) for multiple target tracking. We also discuss several other discoveries and contributions made while pursuing this goal. We begin by defining the problem and discussing why it is important.

1.1 Multiple Target Tracking

The multiple target tracking (MTT) problem is best understood through a simple example, as illustrated in Figure 1. Suppose there is a single sensor, such as a radar, that makes observations of targets, such as airplanes. Now suppose that the sensor detects two targets in the same general area. The first time this happens, the targets are named "Target 1" and "Target 2," respectively. As long as the targets are sufficiently separated, the tracker can easily associate observations with targets. However, when the targets are closely spaced, the tracker makes two observations, A and B, but does not know which observation corresponds to which target. The problem is further complicated by the possibility of missed detections

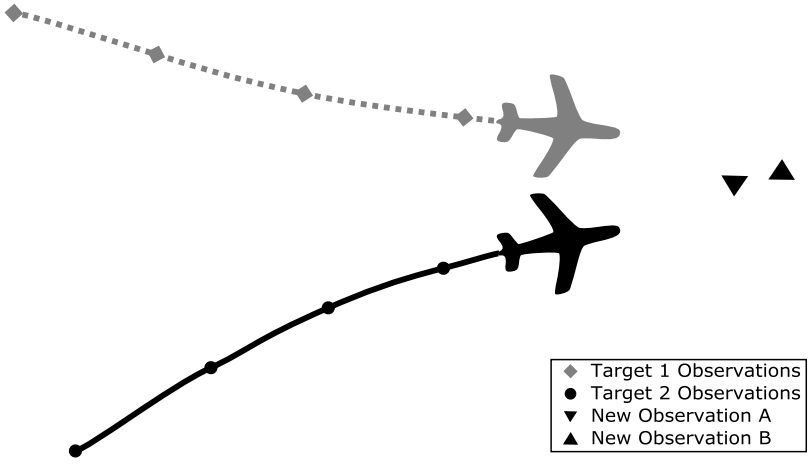


Figure 1: Data association illustration.

and false alarms. Often referred to as the data association problem, this difficulty lies at the heart of multiple target tracking.

A number of algorithms apply probabilistic techniques to the data association problem. One popular approach to MTT is the joint probabilistic data association (JPDA) filter [3, 5], which assigns probabilities to each possible association. Another large family of solutions fall under the name of multiple hypothesis tracking (MHT) [3,5]. MHT finds some number of possible association hypotheses at each time step and maintains these over time to determine which was the most likely. MHT can be implemented in a variety of different ways, but the general idea is to try many possible associations and let time determine which of the hypotheses was most likely. These algorithms perform explicit measurement-to-track associations.

Several techniques have appeared that can be described as "associationless" because they never perform explicit report-to-track associations. One of these is Kastella's event averaged-maximum likelihood estimation (EAMLE), which uses a multidimensional Gaussian probability density function for report-to-track association [22,23]. Another approach is the probability hypothesis density (PHD) filter proposed by Mahler [28,29,39]. The PHD offers a theoretically rigorous (yet approximate) approach to multiple-target, multiple-sensor tracking that naturally estimates the (potentially varying) number of targets. This work centers around another associationless approach to MTT, namely symmetric measurement equations (SME).

1.2 Importance

Optimal solutions to MTT problems are intractable in most situations of interest. Although many suboptimal solutions to MTT problems have been proposed, system designers have few tools at their disposal, other than trial-and-error, for predicting the performance of MTT systems. One common solution to similar challenges has been to develop bounds on the performance of a filtering system, such as the Cramér-Rao lower bound. It turns out that even bounds are difficult to compute for MTT problems. The exponential growth in the number of data association possibilities that plagues attempts at optimal MTT tracking

solutions also plagues attempts to compute bounds. Using an unusual approach to MTT, symmetric measurement equations (SME), this thesis provides methods for computing approximate CRLBs for the MTT problem.

1.3 Cramér-Rao Lower Bounds

1.3.1 CRLB for Scalar Parameters

Often the form of a random variable's probability distribution is known, but the value of a required parameter is not.[1] In this situation, it may be practical to estimate the missing parameter from repeated observations of the random variable. Suppose that $\mathbf{z}$ is a random data vector of length n dependent on the unknown nonrandom parameter x. If an estimator of x, $T(\mathbf{z})$, has the property that

$$E[T(\mathbf{z})] = x \quad \text{for all } x, \tag{1}$$

then it has the highly desirable quality of being unbiased. Now suppose that there are two unbiased estimators, $T_1(\mathbf{z})$ and $T_2(\mathbf{z})$. It is reasonable to assume that the estimator with the lower variance is more desirable. The Cramér-Rao lower bound (CRLB) answers the question, "What is the lowest possible variance for an unbiased estimator?" In some cases, this may be an achievable lower bound [34]. The expression for the CRLB is

$$\begin{aligned} \operatorname{var}(T(\mathbf{z})) &\geq J^{-1}(x), \\ J(x) &= E\left[\left(\frac{\partial}{\partial x}\ln p(\mathbf{z};x)\right)^2\right], \\ J(x) &= -E\left[\frac{\partial^2 \ln p(\mathbf{z};x)}{\partial x^2}\right], \end{aligned} \tag{2}$$

where p represents the probability density function of $\mathbf{z}$ and J is known as the Fisher information [40].

1.3.2 CRLB for Vector Parameters

The CRLB can be extended to include the case of $\mathbf{x}$ being an $r \times 1$ vector parameter rather than a scalar. In this case, the data $\mathbf{z}$ is an $k \times n$ random variable, and the CRLB is given

[1] One example would be estimating the mean (unknown parameter) of a Gaussian random variable from repeated observations.

by

$$\text{var}(T(\mathbf{z})) \geq J^{-1}(\mathbf{x}), \tag{3}$$

$$J(\mathbf{x}) = E\left[(\nabla_{\mathbf{x}} \ln p(\mathbf{z};\mathbf{x}))(\nabla_{\mathbf{x}} \ln p(\mathbf{z};\mathbf{x}))^T\right], \tag{4}$$

$$J(\mathbf{x}) = -E\left[\nabla_{\mathbf{x}}\{(\nabla_{\mathbf{x}} \ln p(\mathbf{z};\mathbf{x}))^T\}\right]. \tag{5}$$

Keep in mind, at this stage, that $\mathbf{z}$ is a random vector, but $\mathbf{x}$ is a deterministic vector.

1.3.3 CRLB for Deterministic Single Target Tracking

The CRLB for single target tracking follows from the vector parameter case. The estimator is now the tracker and the variance of interest is the estimation error covariance (or mean-squared error (MSE)).[2] We assume that the target moves according to a state model, and the state model tells us exactly how it changes from one time to the next. For example, suppose that the target moves in one dimension, y, according to a discrete, constant-velocity model. The equations for its motion are

$$\begin{aligned} y_{k+1} &= y_k + v_k T, \\ v_{k+1} &= v_k, \end{aligned} \tag{6}$$

where v_k is the velocity (a constant), T is the time step, and the subscript indicates time. Written in state equation form, these equations are

$$\underbrace{\begin{bmatrix} y_{k+1} \\ v_{k+1} \end{bmatrix}}_{\mathbf{x}_{k+1}} = \underbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} y_k \\ v_k \end{bmatrix}}_{\mathbf{x}_k}, \tag{7}$$

$$\mathbf{x}_{k+1} = A\,\mathbf{x}_k. \tag{8}$$

The observations are then related to the state by the equation

$$\mathbf{z}_k = g(\mathbf{x}_k, n_k), \tag{9}$$

where g is the observation function and n_k is the observation noise random process. Now the parameter we are trying to estimate is the sequence

$$\mathbf{X}_k = (\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k), \tag{10}$$

[2] We are assuming an unbiased tracker.

from the sequence of observations

$$\mathbf{Z}_k = (\mathbf{z}_0, \mathbf{z}_1, \cdots, \mathbf{z}_k). \tag{11}$$

Substituting $\mathbf{X}_k$ for $\mathbf{x}$ and $\mathbf{Z}_k$ for $\mathbf{z}$ in (3-5), we find the CRLB for the set of deterministic states $\mathbf{X}_k$ at time step k is

$$\text{var}(T(\mathbf{Z}_k)) \geq J^{-1}(\mathbf{X}_k), \tag{12}$$

$$J(\mathbf{X}_k) = E\left[(\nabla_{\mathbf{x}} \ln p(\mathbf{Z}_k; \mathbf{X}_k))(\nabla_{\mathbf{x}} \ln p(\mathbf{Z}_k; \mathbf{X}_k))^T | \mathbf{X}_k\right], \tag{13}$$

$$J(\mathbf{X}_k) = -E\left[\nabla_{\mathbf{x}}\{(\nabla_{\mathbf{x}} \ln p(\mathbf{Z}_k; \mathbf{X}_k))^T\} | \mathbf{X}_k\right]. \tag{14}$$

As k increases, the sizes of $\mathbf{Z}_k$ and $\mathbf{X}_k$ grow rapidly. Fortunately, using our state model setup, the CRLB can be computed recursively. In [36], Taylor provides a method for recursively computing the CRLB for this problem under the assumption that the discrete (potentially nonlinear) observations are corrupted by additive Gaussian white noise. Taylor observed that by constantly replacing an EKF's estimate of the state $\hat{\mathbf{x}}_k$ with the true $\mathbf{x}_k$, the tweaked EKF's state covariance matrix corresponds to the CRLB.

1.3.4 Posterior Cramér-Rao Lower Bound for Single Target Tracking

So far, we have assumed that the target moves according to a deterministic model, which is generally unrealistic. To be practical, (7) should be modified to include a process noise term, $\mathbf{w}_k$:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \mathbf{w}_k. \tag{15}$$

Once the noise term is included, the model is no longer deterministic and the previous equations no longer describe the CRLB. Van Trees addressed this situation by deriving a new bound that is often referred to as the Van Trees bound or the posterior Cramér-Rao bound (PCRB) [40]. Let $T(\mathbf{z}_k)$ be an estimator of the $r \times 1$-dimensional state, $\mathbf{x}_k$. The PCRB is then given by

$$\text{var}(T(\mathbf{Z}_k)) \geq J^{-1}, \tag{16}$$

$$J = E\left[(\nabla_{\mathbf{x}} \ln p(\mathbf{Z}_k, \mathbf{X}_k))(\nabla_{\mathbf{x}} \ln p(\mathbf{Z}_k, \mathbf{X}_k))^T\right], \tag{17}$$

$$J = -E\left[\nabla_{\mathbf{x}}\{(\nabla_{\mathbf{x}} \ln p(\mathbf{Z}_k, \mathbf{X}_k))^T\}\right], \tag{18}$$

where

$$\mathbf{X}_k = (\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k), \tag{19}$$

$$\mathbf{Z}_k = (\mathbf{z}_0, \mathbf{z}_1, \cdots, \mathbf{z}_k). \tag{20}$$

The key difference between the PCRB equations and the single-target tracking CRLB in (12-14) is that $\mathbf{X_k}$ is now a random parameter. In the CRLB case, the probability density functions, $p(\cdot)$, were parameterized by the nonrandom values of $\mathbf{X}_k$. In the PCRB case, the probability density functions are joint probability density functions over the random variables $\mathbf{Z}_k$ and $\mathbf{X}_k$. Furthermore, the expectations in the PCRB case are taken over both of these random variables. Much as Taylor found a method for recursively computing the CRLB, Tichavský *et al.* present a recursive method of calculating the PCRB [38]. Taylor,[3] Flanagan, and Uber present practical methods for using this algorithm that generate approximate PCRBs in [37].

1.3.5 Cramér-Rao Lower Bounds for Multiple Target Tracking

The CRLB and PCRB described above define lower bounds for (linear or nonlinear) estimation problems. However, MTT algorithms typically operate in two stages: the first stage addresses data association, and the second stage performs nonlinear filtering/estimation. The optimal solution to the data association problem is usually impractical to find, even for computing bounds. One previous approach to solving data association for computing CRLBs and PCRBs has been to modify existing suboptimal MTT data association algorithms. Daum presents a lower bound calculation based on multiple hypothesis tracking [9]. The difficulty in this approach is its computational complexity; it essentially reduces to performing Monte Carlo runs where one artificially includes the correct association as one of the runs. In [13], Hue *et al.* describe a PCRB using association probabilities and Monte Carlo integration. In our work, however, we have chosen an uncommon method of multiple target tracking, namely the method of symmetric measurement equations (SME) [18, 19], which is unique because it transforms the data association problem into a set of nonlinear equations. Multiple target tracking with the SME is a single-stage nonlinear estimation problem, and

[3] The author of [36] is James Taylor, and the author of [37] is Robert M. Taylor, Jr.

thus, it provides the means for computing the CRLB and PCRB using methods already available.

1.4 Thesis Organization

This thesis is organized as follows. Chapter 2 introduces the SME approach to multiple target tracking and the associated nonlinear filtering problem. This chapter also presents our unexpected results from using the unscented Kalman filter (UKF) as the nonlinear filter in SME tracking. Chapter 3 presents our new implementation of two-dimensional symmetric measurement equations. Chapter 4 introduces a novel approach to three-dimensional SME tracking. Chapter 5 ties together SME-based MTT and existing approaches for computing Cramér-Rao lower bounds on nonlinear estimators to arrive at the goal of this thesis: approximate Cramér-Rao lower bounds for multiple target tracking.

CHAPTER II

NEW FILTERING METHODS FOR ONE-DIMENSIONAL SME TRACKING

In the early 1990's, Kamen and Sastry presented a novel approach to multiple target tracking based on symmetric measurement equations (SME) [18, 19]. The underlying idea was to create a "pseudomeasurement" vector $\mathbf{s}$ consisting of symmetric functions of the original data $\mathbf{m}$. For example, consider a simple case of tracking three targets in one dimension. Two possible sets of SME's are the *sum of products* and *sum of powers*[1] as follows:

$$\mathbf{s}_{prod} = \begin{pmatrix} m_1 + m_2 + m_3 \\ m_1m_2 + m_1m_3 + m_2m_3 \\ m_1m_2m_3 \end{pmatrix}, \tag{21}$$

$$\mathbf{s}_{pow} = \begin{pmatrix} m_1 + m_2 + m_3 \\ m_1^2 + m_2^2 + m_3^2 \\ m_1^3 + m_2^3 + m_3^3 \end{pmatrix}. \tag{22}$$

Notice that the original m_i's may be rearranged without affecting $\mathbf{s}$. It can be shown that the m_i's may be recovered uniquely (up to a permutation) from $\mathbf{s}$, so there is no fundamental loss of information. This approach turns the data association problem into an analytic nonlinearity. In this way, one difficult problem is traded for another difficult, but quite different, problem.

The first studies of the SME approach used extended Kalman filters (EKF) to handle the nonlinearities. In practice, this turned out to have some difficulties. In this situation, the EKF often exhibits instability, particularly when targets cross, and can be extremely sensitive to initial state and error covariance values as well as the chosen process noise covariance. The EKF is based on first-order Taylor series expansions (linearizations) around the state estimate, and the accuracy of these expansions breaks down if the estimated state

[1] In practice, we subtract some constants to ensure the pseudomeasurements are zero mean. We suppress those constants here to avoid cluttering the exposition.

is too far from the true state. The EKF seems well-suited to handle gentle nonlinearities, but not the sort that arise in the SME approach.

This chapter revisits the SME approach, replacing the extended Kalman filter with the unscented Kalman filter (UKF) and the particle filter. The UKF experiment produced two unexpected results. First, the UKF appears to be fundamentally incompatible with the sum-of-products form of the SME. We show this to be a limitation of the UKF rather than a flaw in the SME approach. Second, we show that the sum-of-powers form, when paired with the UKF, performs as well as the sum-of-products form paired with the EKF. This contrasts with Kamen's early studies where the sum-of-products form outperformed the sum-of-powers form with the EKF; hence, all later work by him and his colleagues focused on the sum-of-products form.

The next step performed in this study was to apply particle filters to the SME approach. The results from studying the EKF, the UKF, and the particle filter suggest that the performance of the SME approach is extremely dependent on the pairing of an SME implementation and nonlinear filter, rather than dependent on either individually.

2.1 The Unscented Kalman Filter

2.1.1 The Unscented Transform

The unscented Kalman filter relies upon a mathematical technique referred to as the unscented transform [16]. The unscented transform captures the mean and variance of any random variable using a set of "sigma points." The sample mean and variance of these sigma points match the statistics of the original random variable.

Julier and Uhlmann show, in [17], that finding these sigma points is a straightforward process based upon the square root of the random variable's covariance matrix. While any square root is acceptable, the Cholesky decomposition is generally chosen for its numerical stability.

For example, let $\mathbf{x}$ be an n-dimensional random variable with mean μ and covariance matrix $P_{\mathbf{xx}}$. Let $\mathcal{X}$ be the set of sigma points for the random variable $\mathbf{x}$. Julier and Uhlmann

show, in [16], that the $2n+1$ sigma points should be chosen as follows:

$$\mathcal{X}_0 = \mu, \tag{23}$$

$$\sigma = 2n \text{ columns of } \pm\sqrt{(n+\lambda)P_{\mathbf{xx}}}, \tag{24}$$

$$\mathcal{X}_i = \mu \ + \ \sigma_i, \tag{25}$$

where

$$(n+\lambda)P_{\mathbf{xx}} = \sqrt{(n+\lambda)P_{\mathbf{xx}}}^T \sqrt{(n+\lambda)P_{\mathbf{xx}}}, \tag{26}$$

and $\lambda = \alpha^2(n+\kappa) - n$ is a composite scaling parameter. α controls the spread of the sigma points around the mean, and we have $\alpha = 1$ from the recommended range $0.0001 \leq \alpha \leq 1$ [41]. The authors of [16] recommend choosing κ such that $n + \kappa = 3$, unless this leads to $\kappa < 0$,[2] in which case choose $\kappa = 0$. Since $n = 4$ is the minimum value of n considered in this dissertation, we set $\kappa = 0$. This choice of α and κ leads to $\lambda = 0$ for our simulations.

Suppose we are interested in the random variable $\mathbf{y} = g(\mathbf{x})$,[3] $g(\cdot)$ is any function of an n-dimensional random variable. The estimated statistics of $\mathbf{y}$ can be found by generating a new set of sigma points, $\mathcal{Y}$, where

$$\mathcal{Y}_i = g(\mathcal{X}_i) \qquad \text{for i} = 0,\ldots,2n. \tag{27}$$

The estimated statistics of $\mathbf{y}$ are then related to the sample mean and variance of $\mathcal{Y}$.

$$E[\mathbf{y}] \approx \sum_{i=0}^{2n} W_i^{(m)} \mathcal{Y}_i, \tag{28}$$

$$P_{\mathbf{yy}} = E\left[(\mathbf{y} - E[\mathbf{y}])(\mathbf{y} - E[\mathbf{y}])^T\right] \approx \sum_{i=0}^{2n} W_i^{(c)} [\mathcal{Y}_i - \mathbf{y}][\mathcal{Y}_i - \mathbf{y}]^T, \tag{29}$$

where

$$W_0^{(m)} = \lambda/(n+\lambda), \tag{30}$$

$$W_0^{(c)} = \lambda/(n+\lambda) + (1 - \alpha^2 + \beta), \tag{31}$$

$$W_i^{(m)} = W_i^{(c)} = 1/(2(n+\lambda)) \quad i = 1,\ldots,2n. \tag{32}$$

[2] Choosing $\kappa < 0$ may lead to non-positive semidefinite estimates of the posterior covariance matrix.

[3] We typically use $\mathbf{s}$ as the output of $g(\cdot)$ because $g(\cdot)$ is usually an SME in this dissertation. However, we have used $\mathbf{y}$ here and several other places where it is customary to do so in the UKF literature.

β is a parameter that can be used to incorporate knowledge about the distribution of $\mathbf{x}$. $\beta = 2$ is the optimal choice for the Gaussian distributions of $\mathbf{x}$ used in this dissertation. These estimated statistics are accurate up to a third-order Taylor series expansion [41]. Figure 2 gives an illustration of the unscented transform for $\mathbf{y} = g(\mathbf{x})$, where $\mathbf{x}$ and $\mathbf{y}$ are two-dimensional random variables.

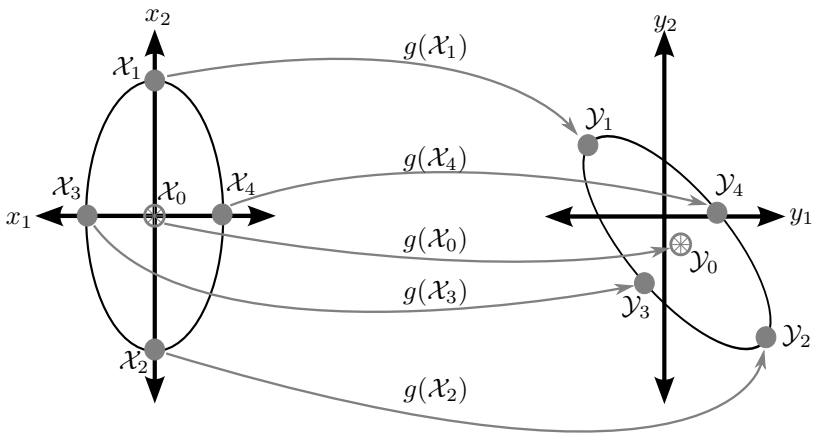


Figure 2: Illustration of the unscented transform.

2.1.2 Building a Kalman Filter

Here we present a brief overview of the UKF; see [16] and [41] for a detailed derivation. Building the unscented transform into a filter requires augmenting the state variable to include noise terms. Specifically, we now define our state at time instant k as

$$\mathbf{x}_k^a = \begin{pmatrix} \mathbf{x}_k \\ \mathbf{w}_k \\ \mathbf{v}_k \end{pmatrix}, \tag{33}$$

where $\mathbf{x}_k$ is the original (desired) state information, $\mathbf{w}_k$ is the process noise, and $\mathbf{v}_k$ is the observation noise. The covariance, $P_{\mathbf{x}_k^a \mathbf{x}_k^a}$, is

$$P_{\mathbf{x}_k^a \mathbf{x}_k^a} = \begin{pmatrix} P_{\mathbf{xx}} & P_{\mathbf{xw}} & P_{\mathbf{xv}} \\ P_{\mathbf{xw}} & P_{\mathbf{ww}} & P_{\mathbf{wv}} \\ P_{\mathbf{xv}} & P_{\mathbf{wv}} & P_{\mathbf{vv}} \end{pmatrix}. \tag{34}$$

The unscented transform is then applied to the augmented state, $\mathbf{x}_k^a$, and augmented covariance, $\mathbf{x}_k^a$ and $P_{\mathbf{x}_k^a \mathbf{x}_k^a}$, to generate a set of sigma points, $\mathcal{X}_k$. To find the predicted state, the state transition function, $f(\cdot)$, is applied to the sigma points $\mathcal{X}_k$, generating a new set

of sigma points $\mathcal{X}_{k+1|k}$. The predicted state, $\mathbf{x}^-_{k+1}$, and the predicted covariance, $P^-_{\mathbf{x}^a_{k+1}\mathbf{x}^a_{k+1}}$, are the weighted sample statistics of $\mathcal{X}_{k+1|k}$ given by

$$\mathbf{x}^-_{k+1} = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{X}_{i,k+1|k}, \tag{35}$$

$$P^-_{\mathbf{x}^a_{k+1}\mathbf{x}^a_{k+1}} = \sum_{i=0}^{2n} W_i^{(c)} [\mathcal{X}_{i,k+1|k} - \mathbf{x}^-_{k+1}][\mathcal{X}_{i,k+1|k} - \mathbf{x}^-_{k+1}]^T. \tag{36}$$

Now suppose $g(\cdot)$ is the observation function. A third set of sigma points, $\mathcal{Y}_{k+1|k}$, are found to represent the predicted observation

$$\mathcal{Y}_{k+1|k} = g(\mathcal{X}_{k+1|k}). \tag{37}$$

The predicted observation, $\mathbf{y}^-_{k+1}$, and the predicted observation covariance, $P_{\mathbf{y}_{k+1}\mathbf{y}_{k+1}}$, are the weighted sample statistics of $\mathcal{Y}_{k+1|k}$ given by

$$\mathbf{y}^-_{k+1} = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{Y}_{i,k+1|k}, \tag{38}$$

$$P_{\mathbf{y}_{k+1}\mathbf{y}_{k+1}} = \sum_{i=0}^{2n} W_i^{(c)} [\mathcal{Y}_{i,k+1|k} - \mathbf{y}^-_{k+1}][\mathcal{Y}_{i,k+1|k} - \mathbf{y}^-_{k+1}]^T. \tag{39}$$

The predicted cross correlation, $P_{\mathbf{x}^a_{k+1}\mathbf{y}_{k+1}}$, is the sample cross correlation of $\mathcal{X}_{k+1|k}$ and $\mathcal{Y}_{k+1|k}$

$$P_{\mathbf{x}^a_{k+1}\mathbf{y}_{k+1}} = \sum_{i=0}^{2n} W_i^{(c)} [\mathcal{X}_{i,k+1|k} - \mathbf{x}^-_{k+1}][\mathcal{Y}_{i,k+1|k} - \mathbf{y}^-_{k+1}]^T. \tag{40}$$

We now have the familiar setup of a Kalman filter. The Kalman gain, K, is

$$K = P_{\mathbf{x}^a_{k+1}\mathbf{y}_{k+1}} P^{-1}_{\mathbf{y}_{k+1}\mathbf{y}_{k+1}}. \tag{41}$$

The filter estimate of the state is then

$$\mathbf{x}^a_{k+1} = \mathbf{x}^-_{k+1} + K(\mathbf{y}_{k+1} - \mathbf{y}^-_{k+1}), \tag{42}$$

$$P_{\mathbf{x}^a_{k+1}\mathbf{x}^a_{k+1}} = P^-_{\mathbf{x}^a_{k+1}\mathbf{x}^a_{k+1}} - K P_{\mathbf{y}_{k+1}\mathbf{y}_{k+1}} K^T. \tag{43}$$

2.1.3 A Limitation of the Unscented Kalman Filter

This section uses a simple example, the sum-of-products SME, to illustrate a limitation of the UKF. Suppose we are tracking two targets in one-dimensional space. Define the state to be

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \\ x_2 \end{pmatrix}, \tag{44}$$

where x_1 and x_2 are the positions of targets one and two. Note, for simplification, we have not included the velocity terms of a constant velocity model or augmented the state vector with noise parameters, as is necessary for the actual UKF. This simplifies analysis without changing the conclusion. Completing the same analysis with the velocity terms and augmented state vector is straightforward.

Following the unscented transform, we need $2n + 1 = 5$ sigma points. Let these sigma points be

$$\mathcal{X}_k = \begin{bmatrix} x_1 & x_1+p_{11} & x_1-p_{11} & x_1+p_{12} & x_1-p_{12} \\ x_2 & x_2+p_{21} & x_2-p_{21} & x_2+p_{22} & x_2-p_{22} \end{bmatrix}, \tag{45}$$

where

$$\sqrt{(n+\lambda)P_{\mathbf{xx}}} = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}. \tag{46}$$

Note that (46) is written as a full matrix since the UKF allows any form of the matrix square root, although the triangular Cholesky decomposition is the one typically used. Applying the state transition function $f(\cdot)$ to $\mathcal{X}_k$, the set of predicted[4] sigma points in the state space is

$$\mathcal{X}_{k+1|k} = \begin{bmatrix} x_1^- & x_1^-+p_{11} & x_1^--p_{11} & x_1^-+p_{12} & x_1^--p_{12} \\ x_2^- & x_2^-+p_{21} & x_2^--p_{21} & x_2^-+p_{22} & x_2^--p_{22} \end{bmatrix}. \tag{47}$$

The sum-of-products observation function is

$$\mathbf{y} = g(\mathbf{x}) = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ x_1 x_2 \end{pmatrix}, \tag{48}$$

so the set of predicted sigma points becomes

$$\mathcal{Y}^T_{k+1|k} = \begin{bmatrix} y_1^- & y_2^- \\ y_1^-+p_{11}+p_{21} & y_2^-+x_1^-p_{21}+x_2^-p_{11}+p_{11}p_{21} \\ y_1^--p_{11}-p_{21} & y_2^--x_1^-p_{21}-x_2^-p_{11}+p_{11}p_{21} \\ y_1^-+p_{12}+p_{22} & y_2^-+x_1^-p_{22}+x_2^-p_{12}+p_{12}p_{22} \\ y_1^--p_{12}-p_{22} & y_2^--x_1^-p_{22}-x_2^-p_{12}+p_{12}p_{22} \end{bmatrix}. \tag{49}$$

The interesting case occurs when the target estimates cross paths and $x_1^- = x_2^-$. The

[4]The superscript $^-$ indicates a value after the state transition function has been applied, but before the observation at the current time step has incorporated.

predicted sigma points become

$$\mathcal{Y}_{k+1|k}^{T} = \begin{bmatrix} y_1^- & y_2^- \\ y_1^- + p_{11} + p_{21} & y_2^- + x_1^-(p_{21}+p_{11}) + p_{11}p_{21} \\ y_1^- - p_{11} - p_{21} & y_2^- - x_1^-(p_{21}+p_{11}) + p_{11}p_{21} \\ y_1^- + p_{12} + p_{22} & y_2^- + x_1^-(p_{22}+p_{12}) + p_{12}p_{22} \\ y_1^- - p_{12} - p_{22} & y_2^- - x_1^-(p_{22}+p_{12}) + p_{12}p_{22} \end{bmatrix}. \tag{50}$$

One final observation is required to understand when the UKF breaks down. When the targets behave independently,[5] the cross-correlation terms of $(n+\lambda)P_{\mathbf{xx}}$ shrink towards zero as the filter locks on to the targets' behavior. In equation form,

$$(n+\lambda)P_{\mathbf{xx}} \rightarrow \begin{pmatrix} p_{11}^2 & 0 \\ 0 & p_{22}^2 \end{pmatrix}. \tag{51}$$

Recall (46), where we defined $\sqrt{(n+\lambda)P_{\mathbf{xx}}}$ such that

$$(n+\lambda)P_{\mathbf{xx}} = \begin{pmatrix} p_{11}\ p_{21} \\ p_{12}\ p_{22} \end{pmatrix} \begin{pmatrix} p_{11}\ p_{12} \\ p_{21}\ p_{22} \end{pmatrix} \tag{52}$$

$$= \begin{pmatrix} p_{11}^2 + p_{21}^2 & p_{11}p_{12} + p_{21}p_{22} \\ p_{11}p_{12} + p_{21}p_{22} & p_{22}^2 + p_{12}^2 \end{pmatrix}. \tag{53}$$

Comparing (51) and (53), p_{12} and p_{21} tend towards zero when our assumption that the targets move independently is met. This leaves our predicted sigma points in the following form

$$\mathcal{Y}_{k+1|k}^{T} = \begin{bmatrix} y_1^- & y_2^- \\ y_1^- + p_{11} & y_2^- + x_1 p_{11} \\ y_1^- - p_{11} & y_2^- - x_1 p_{11} \\ y_1^- + p_{22} & y_2^- + x_1 p_{22} \\ y_1^- - p_{22} & y_2^- - x_1 p_{22} \end{bmatrix}. \tag{54}$$

The covariance matrix is calculated as

$$P_{\mathbf{yy}} = \sum_{i=0}^{2n} W_i^{(c)} [\mathcal{Y}_{i,k+1|k} - \mathbf{y}][\mathcal{Y}_{i,k+1|k} - \mathbf{y}]^T, \tag{55}$$

[5]Targets moving independently, by definition, have no correlation in their behavior. The filter models this by estimating a state covariance matrix with small off-diagonal terms.

where the $\mathcal{Y}_{i,k+1|k}$ are the individual sigma points of $\mathcal{Y}_{k+1|k}$. Using the sigma points in (54), setting $e_i = y_i^- - y_i$, and computing $P_{\mathbf{yy}}$ according to (55) leads to

$$P_{\mathbf{yy}} = \begin{bmatrix} ce_1^2 + d(p_{11}^2 + p_{22}^2) & ce_1e_2 + dx_1(p_{11}^2 + p_{22}^2) \\ ce_1e_2 + dx_1(p_{11}^2 + p_{22}^2) & ce_2^2 + dx_1^2(p_{11}^2 + p_{22}^2) \end{bmatrix}, \tag{56}$$

where

$$c = W_0^{(c)} + 4W_i^{(c)}, \tag{57}$$

$$d = 2W_i^{(c)}. \tag{58}$$

Since e_i is the difference of the predicted observation and the actual observation, it can be decomposed into two parts: the prediction error and the observation noise such that

$$e_i = e_{i,pred} + v_i. \tag{59}$$

In the extreme case that $e_1 = e_2 = 0$, $P_{\mathbf{yy}}$ becomes a singular matrix with the second column being the first scaled by x_1. This means that the previously two-dimensional space has degenerated into a one-dimensional space – all the sigma points lie along a single line as seen in the top, right portion of Figure 3. The reduction in dimensionality means the covariance matrix $P_{\mathbf{yy}}$ is ill-conditioned and the inverse, if it exists at all, will have elements with extremely large values. The Kalman gain, $K = P_{\mathbf{xy}}P_{\mathbf{yy}}^{-1}$, will then multiply the observed data by enormous values and produce inaccurate target tracks. For the matrix inversion to be ill-conditioned, it is sufficient for the e_i to be small instead of exactly zero. Since the prediction error $e_{i,pred}$ is likely to be small if the tracker is performing well and since v_i is zero mean, it is likely that the e_i's will be small and this problem will occur. It should also be noted that these results are independent of the choice of the UKF scaling parameters α, β, κ, and λ since the loss of dimensionality does not depend upon their values. Indeed, this is an example of a weakness in the UKF formulation that is already known, but often overlooked [44]. The UKF only guarantees that $P_{\mathbf{yy}}$ will be positive semidefinite. In the situation described above, $P_{\mathbf{yy}}$ will have zero or near-zero eigenvalues.

The need to artificially increase the covariance for numerical stability has been addressed in [15], [25]. A simple solution would be to set a minimum value for the covariance matrix.

Experimental results show that this approach reduces, but does not remove, the degeneration of $P_{\mathbf{yy}}$ and may still lead to problems maintaining its positive-definiteness. This is, however, simply modifying the UKF with ad-hoc techniques, which defeats the purpose of using it in the first place.

One potential alternative to the UKF is the divided difference filter (DDF) [32] (also known as the central difference filter [14]). However, analysis and experimentation with the DDF demonstrated that it suffers the same singularity problem as the UKF, which is no surprise given the similarity between the two filters.

2.1.4 When the Unscented Kalman Filter Works

Section 2.1.3 showed how the UKF breaks down when paired with the sum-of-products form of the SME. This section follows a similar derivation for the sum-of-powers SME implementation and illustrates how the problems of the previous section are avoided by the sum of powers.

The setup is the same as in Section 2.1.3, with

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{60}$$

and

$$\mathcal{X}_{k+1|k} = \begin{bmatrix} x_1^- & x_1^- + p_{11} & x_1^- - p_{11} & x_1^- + p_{12} & x_1^- - p_{12} \\ x_2^- & x_2^- + p_{21} & x_2^- - p_{21} & x_2^- + p_{22} & x_2^- - p_{22} \end{bmatrix}. \tag{61}$$

Instead of (48), the observation function is

$$g(\mathbf{x}) = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ x_1^2 + x_2^2 \end{pmatrix}, \tag{62}$$

which leads to the following set of predicted sigma points

$$\mathcal{Y}_{k+1|k}^T = \begin{bmatrix} y_1^- & y_2^- \\ y_1^- + p_{11} + p_{21} & y_2^- + 2x_1p_{11} + 2x_2p_{21} + p_{11}^2 + p_{21}^2 \\ y_1^- - p_{11} - p_{21} & y_2^- - 2x_1p_{11} - 2x_2p_{21} + p_{11}^2 + p_{21}^2 \\ y_1^- + p_{12} + p_{22} & y_2^- + 2x_1p_{12} + 2x_2p_{22} + p_{12}^2 + p_{22}^2 \\ y_1^- - p_{12} - p_{22} & y_2^- - 2x_1p_{12} - 2x_2p_{22} + p_{12}^2 + p_{22}^2 \end{bmatrix}. \tag{63}$$

Again examining the case of target estimates crossing and assuming that p_{12} and p_{21} are nearly zero, the predicted sigma points become

$$\mathcal{Y}_{k+1|k}^{T} = \begin{bmatrix} y_1^- & y_2^- \\ y_1^- + p_{11} & y_2^- + 2x_1 p_{11} + p_{11}^2 \\ y_1^- - p_{11} & y_2^- - 2x_1 p_{11} + p_{11}^2 \\ y_1^- + p_{22} & y_2^- + 2x_1 p_{22} + p_{22}^2 \\ y_1^- - p_{22} & y_2^- - 2x_1 p_{22} + p_{22}^2 \end{bmatrix}. \tag{64}$$

Even in the extreme case that $e_1 = e_2 = 0$, using the sigma points in (64) and applying (55) leads to

$$P_{\mathbf{yy}} = \begin{bmatrix} 2W_i^{(c)}(p_{11}^2 + p_{22}^2) & 4W_i^{(c)} x_1 (p_{11}^2 + p_{22}^2) \\ 4W_i^{(c)} x_1 (p_{11}^2 + p_{22}^2) & 8W_i^{(c)} x_1^2 (p_{11}^2 + p_{22}^2) + 2W_i^{(c)} (p_{11}^4 + p_{22}^4) \end{bmatrix}. \tag{65}$$

In contrast to Section 2.1.3, the columns of $P_{\mathbf{yy}}$ are clearly not linearly dependent. Consequently, $P_{\mathbf{yy}}$ remains full rank and its inverse is well-defined. The lower right portion of Figure 3 shows the graphical perspective. Again, all the sigma points fall onto the same curve, but this time the curve is quadratic, maintaining two dimensions rather than collapsing to one, as with the sum of products.

2.2 The Particle Filter

The particle filter implementation was chosen to provide a bound on the performance of the EKF and UKF. Since the particle filter allows the posterior to be non-Gaussian, we expected it to have the best performance [11]. In practice, however, finding the exact likelihood function for the particle filter is difficult because the highly nonlinear SME transformations produce difficult-to-calculate measurement noise densities. Rather than calculate the noise density exactly, the particle filter was implemented with two different additive Gaussian noise approximations. The first is an analytically derived approximation from the EKF [20], and the second is based on the unscented transform discussed in Section 2.1.1.

As implemented in this work, each particle contains an entry for every element of state information. In the two-target one-dimensional case, for example, there are four elements

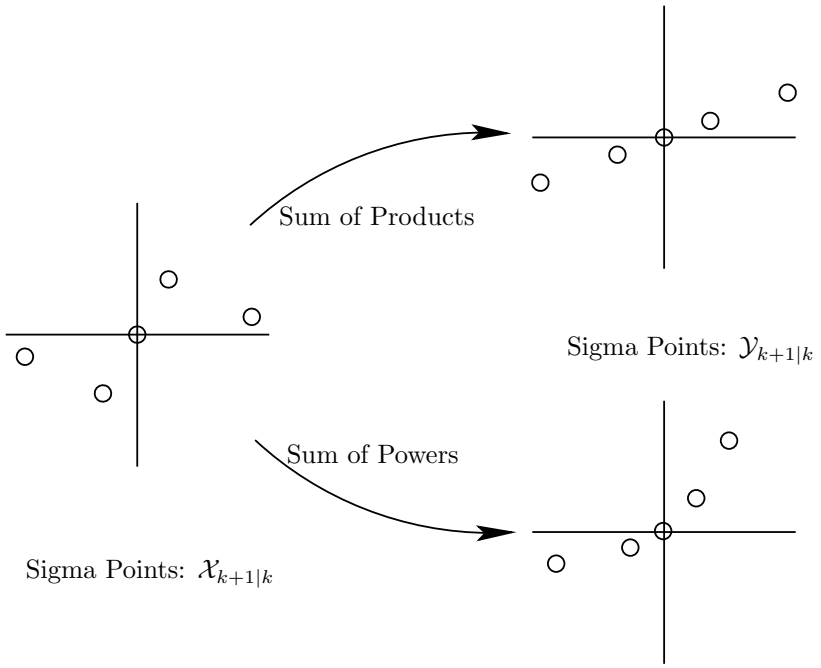


Figure 3: Illustration of the sigma point transformation for the sum-of-products and sum-of-powers SMEs.

in the state: two positions and two velocities. The set of predicted particles is found by applying a constant velocity model to the existing particles. To evaluate the likelihood of each predicted particle, the SME equations are applied to the position elements of that particle to generate the particle's SME "position." The likelihood of the SME observation at the current time step is approximated in SME space by evaluating the probability of that observation according to a Gaussian distribution around the particle's SME "position." The following sections discuss two possible methods for finding a suitable Gaussian distribution.

2.2.1 Noise in the SME Formulation

Suppose our system is tracking N targets, and let the actual received measurement, m_i, be modeled as the truth, x_i, plus an additive Gaussian noise, n_i, so that

$$m_i = x_i + n_i \qquad \text{for } i = 1, \ldots, N. \tag{66}$$

Applying an SME observation function $g_i(\cdot)$ to the actual measurements, m_i, yields the SME measurements, s_i,

$$s_i = g_i(m_1, m_2, \cdots, m_N) \quad \text{for } i = 1, \ldots, N. \tag{67}$$

The restrictions on the choice of SME functions given in [18] require that (67) can be written as

$$s_i = g_i(x_1, x_2, \cdots, x_N) + v_i \quad \text{for } i = 1, \ldots, N, \tag{68}$$

where v_i is a zero mean white noise term whose distribution may depend on $x_1, x_2, \cdots, x_N$. Note that v_i is not Gaussian, because the SME functions are nonlinear.

2.2.2 Analytical Gaussian Approximation

Despite the statement in Section 2.2.1 above that the noise terms v_i are not Gaussian, they can be approximated as such. Kamen presents a detailed derivation of an analytical Gaussian approximation for the sum-of-products in [20]. Unfortunately, the derivation for the sum-of-powers case is not nearly as neat; the final result has been included in Appendix A. Furthermore, deriving the approximations becomes increasingly tedious as the number of targets increases. It is this approximation, however, that Kamen *et al.* used with the SME-EKF pairing. The same approximation can be used with a particle filter. The results using this approximation with the particle filter are given in the *Particle Filter - Analytical* sections of Tables 1, 2, and 3.

2.2.3 Unscented Transform-based Gaussian Approximation

Another approach to generating a Gaussian approximation is to use the unscented transform discussed in 2.1.1. The advantage of using the unscented transform is that it tries to approximate the non-Gaussian noise. Consequently, it should produce a better-performing particle filter.

To implement this approach, the unscented transform is applied to each particle of the filter at each time step. This is computationally expensive; however, for our analytical purposes, we do not need the particle filter to run in real-time. Unfortunately, the unscented transform does not work for the sum-of-products formulation for the same reasons that it does not work in the UKF, as discussed in Section 2.1.3. The results in Section 2.3, however, show improved performance in the sum-of-powers case over the analytical Gaussian approximation. Note that this is not the "unscented particle filter" as presented in [41].

The unscented particle filter in [41] uses the UKF to estimate the transition prior density; whereas, we use the unscented transform to provide a better estimate of the likelihood density.

2.3 Results

2.3.1 Simulation Setup

The simulation consisted of three targets moving in one dimension. Each target moved independently with nearly constant velocity and a known process noise variance (10 m^2/s^2). Initial positions and velocities were chosen so that the targets were likely to cross paths. The observation data were generated by adding Gaussian noise with zero mean and a known variance (1×10^4 m^2). Figure 4 gives a sample set of tracks. Time is plotted on the horizontal axis, and position is plotted on the vertical axis. Software developed by Wan and van der

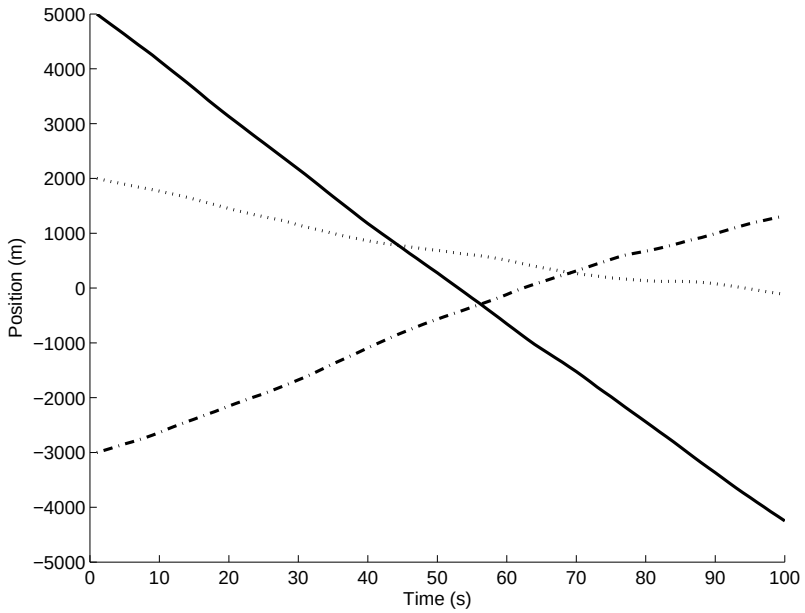


Figure 4: Sample target paths in one-dimension.

Merwe provided the basis for these simulations [42].

To ease discussion, pairing a filter with an SME will be referred to by a pair of abbreviations, e.g. UKF-Products for the unscented Kalman filter paired with the sum of products. We have also included two non-SME data association techniques for reference: the global nearest neighbor (GNN) approach and an associated KF. The GNN algorithm represents

a straightforward solution to the multiple target tracking problem. Since the simulation scenario does not include false alarms and missed detections, each observation is used to update exactly one track. The GNN algorithm associates observations with tracks such that the sum of the distances from the observations to the predicted positions is minimized [5]. This approach may also be thought of as a 2-D assignment algorithm. Separate standard Kalman filters are run for each track. The associated KF, on the other hand, has knowledge of the correct target associations and runs a standard Kalman filter for each track. In our simulation, the associated Kalman filter represents an optimal solution since the data association is perfect and the state model and observation model meet the requirements for the Kalman filter to be the optimal solution. This provides a performance baseline.

2.3.2 Data

Table 1: Percentage of Trials with Correctly Maintained Associations

	Sum of Products	Sum of Powers
Extended Kalman	96	82
Unscented Kalman	58	96
PF - Analytical	88	87
PF - Unscented	−−	95
Global NN	89	
Associated KF	100	

Table 2: Associated Root Mean Squared Error

	Sum of Products	Sum of Powers
Extended Kalman	44.5	45.3
Unscented Kalman	63.7	44.3
PF - Analytical	115.7	97.6
PF - Unscented	−−	105.9
Global NN	42.2	
Associated KF	41.7	

Table 1 contains the percentage of time that each filter maintained the correct target associations for the three targets moving in one-dimensional space. Table 2 contains an associated root mean squared error (RMSE) measurement. The term "associated" indicates

Table 3: Set Estimation Root Mean Squared Error

	Sum of Products	Sum of Powers
Extended Kalman	29.5	30.0
Unscented Kalman	59.8	29.3
PF - Analytical	96.4	81.6
PF - Unscented	−−	72.7
Global NN	28.3	
Associated KF	27.6	

that the error is only included in the average when the filter maintains the correct target association. Table 3 contains the "set estimation" RMSE. The set estimation RMSE is calculated by finding the track-estimate/track-truth association with the smallest RMSE at each time step and then averaging over all time steps. Consequently, target associations are not considered.

Several relationships evident in the tables are worthy of mention. First, the EKF clearly performs better with the sum-of-products implementation (96%) than with the sum of powers (82%). This result was expected since the original investigators of the SME approach used only the EKF and quickly abandoned the sum of powers.

Second, we see the limitation of the UKF with the sum of products (58%) as discussed in Section 2.1.3. The consequences of this issue are illustrated in Figure 5. Notice how the estimated target tracks may jump when two estimates cross.

The most important result is that, when paired with the UKF, the sum-of-powers matches (96%) the best EKF implementation. Not only is this result new and unexpected, but the UKF is significantly easier to implement and has similar computational complexity as the EKF. Also, by both MSE metrics, UKF-Powers has the best performance.

Judging by the percentage of correctly maintained tracks, Punctuated-Powers showed good performance (95%). Data is omitted for Punctuated-Products because the unscented transform is incompatible with the sum-of-products SME for the same reasons the UKF is incompatible with it. By both MSE metrics, the particle filters exhibit poor performance. This result is examined in the following section.

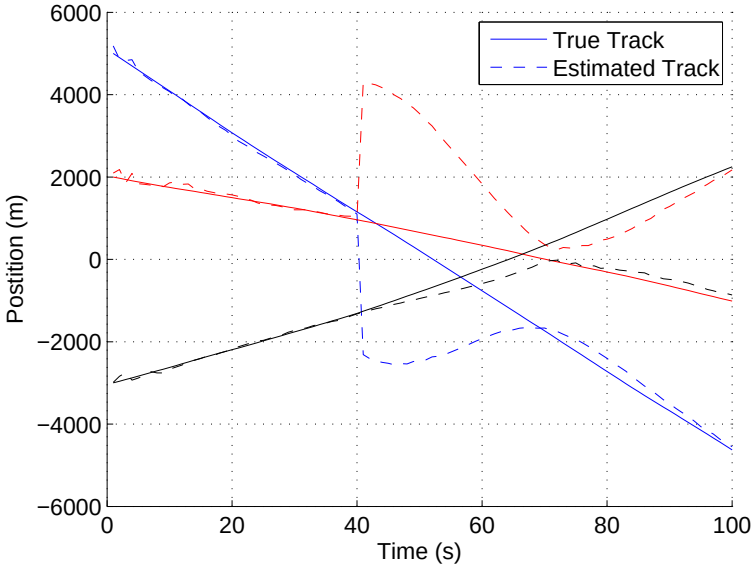


Figure 5: Example of the UKF malfunctioning with the sum-of-products SME as target estimates cross.

2.4 Difficulties with the Particle Filter

2.4.1 Particle Proposal Issue

During our experiments, we noticed that the particle filter would perform decently at lower observation noise levels, but then performance would plummet when the observation noise level became too high. Oddly, raising the process noise would actually cause performance to improve under the high observation noise regime.

The problem stemmed from a mismatch between the location of predicted particles and the location of the observations. The process noise controls the spread of the predicted particles, and the observation noise describes how far an observation might lie from its true, hopefully predicted, position. For the particle filter to work well, some of the particles must be located close to the observation. However, if the observation noise is large compared to the process noise, few, if any, of the predicted particles will be near the observation. This leads to poor evaluation of the particle likelihoods.

To test this explanation, we re-ran the simulation with a lower observation noise (100 m^2) and the same process noise (10 m^2/s^2). Table 4 gives the resulting set error. From these results, it is clear that the particle filters perform much better than before, but they

Table 4: Associated Root Mean Squared Error

	Sum of Products	Sum of Powers
Extended Kalman	6.4	6.1
Unscented Kalman	6.4	6.1
PF - Analytical	6.9	6.5
PF - Unscented	−−	6.5
Global NN	6.1	
Associated KF	6.0	

still trail the Kalman filters.

In general, this problem can be overcome using a more complicated particle proposal method, such as the method presented in [33]. However, we have chosen not to do this since the particle filter has another significant problem, as described in the following section.

2.4.2 RMSE Performance Issue

The poor RMSE performance of the two particle filter implementations can be explained by examining the observation likelihood, $p(\mathbf{s}|\mathbf{x})$, where $\mathbf{s}$ is an SME observation and $\mathbf{x}$ is one of the particles. Our goal is to understand how the particle filter weights particles near an observation. Intuitively, a particle closer to the observation should be more likely according to the likelihood functions. However, the following will show that this is not the case for the particle filters implemented in this chapter.

Recall that the particle filters do not use the true observation likelihood probability density, but instead use a Gaussian approximation to it (2.2.1). One implementation, referred to as the analytical approximation, uses the same approximation as the EKF. The second implementation uses the unscented transform (UT). Imagine two targets moving in one dimension, described by the vector

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \tag{69}$$

Now imagine an SME observation $\mathbf{s} = g_{SME}(\mathbf{x})$. Define a three-dimensional set of axes, as in Figure 6, where the first two dimensions correspond to the positions of targets 1 and 2, x_1 and x_2 respectively, and the third dimension is $p(\mathbf{s}|\mathbf{x})$. To picture $p(\mathbf{s}|\mathbf{x})$, place "particles" in

a uniform grid in the plane defined by x_1 and x_2 axes. Assume that the received observation corresponds to the center of this grid, and calculate the likelihood at each point on the grid.

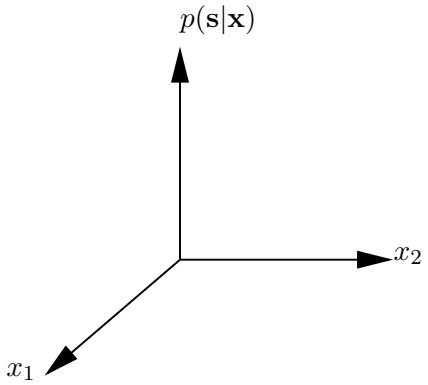


Figure 6: Coordinate axes for visualizing $p(\mathbf{s}|\mathbf{x})$.

For the first example, assume that the measurement, $\mathbf{m}$, which is in the same space as the state,[6] $\mathbf{x}$, is

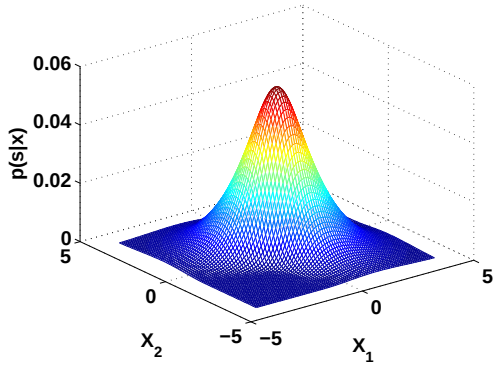
$$\mathbf{m} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{70}$$

in Cartesian coordinates before the SME functions are applied. The "likelihood" for the sum-of-powers is plotted in Figure 7. The unscented transform approximation and the EKF approximation are similar, with the UT approximation having a wider distribution. More importantly, the peaks are located at $(0, 0)$ – the location of the "observation."

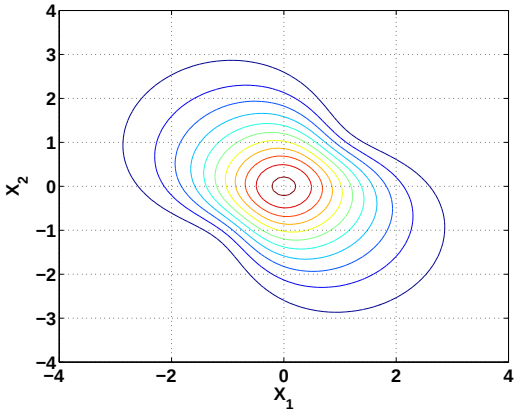
Figure 8 shows the same information, except the observation is now at $(3, 0)$. Note that two peaks of equal magnitude are now visible. This makes sense because the SME approach does not differentiate between $(3, 0)$ and $(0, 3)$. However, the peaks are not located exactly at $(3, 0)$ and $(0, 3)$. Instead, the peaks are shifted towards each other. Consequently, the particles with the highest weights will be consistently offset from the observations, and the particle filter's performance will suffer.

Figure 9 shows the $(3, 0)$ case for the sum-of-products. The UT approximation is not shown because the unscented transform breaks down as described in Chapter 2.1.3. The characteristics observed in the sum-of-powers case also appear in the sum-of-products case.

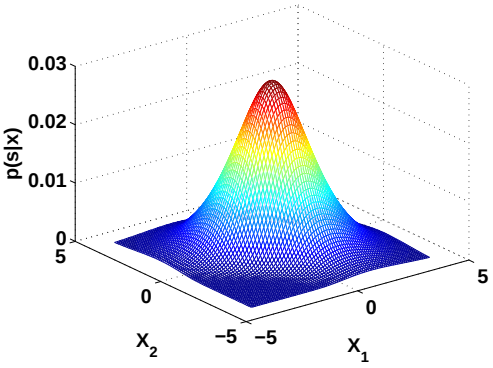
[6]This is not a requirement, but a simplifying assumption that has been made throughout this dissertation.



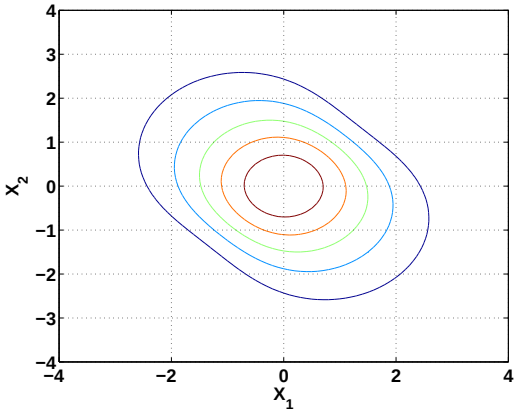
(a) Analytical approximation around $(0, 0)$.



(b) Analytical approximation around $(0, 0)$.

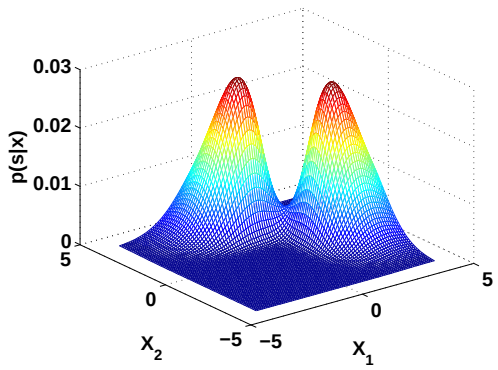


(c) UT approximation around $(0, 0)$.

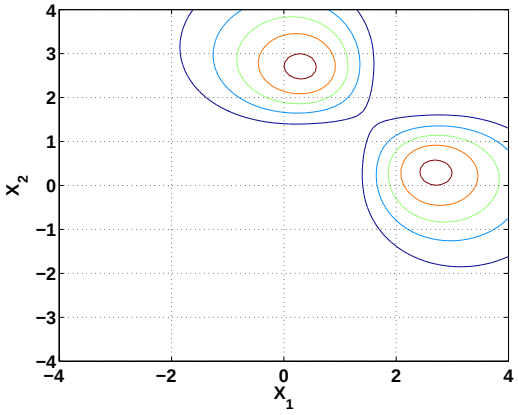


(d) UT approximation around $(0, 0)$.

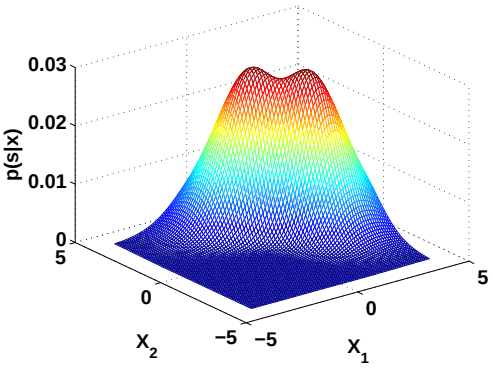
Figure 7: "Likelihood" for two targets moving in one dimension; sum-of-powers SME.



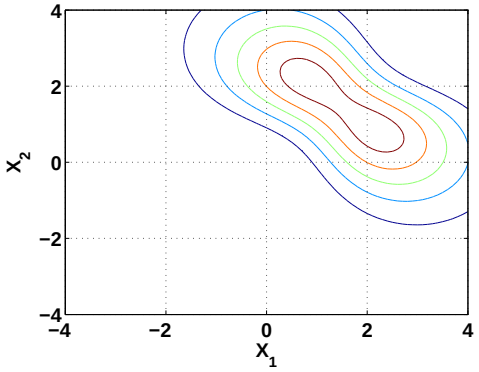
(a) Analytical approximation around $(3, 0)$.



(b) Analytical approximation around $(3, 0)$.

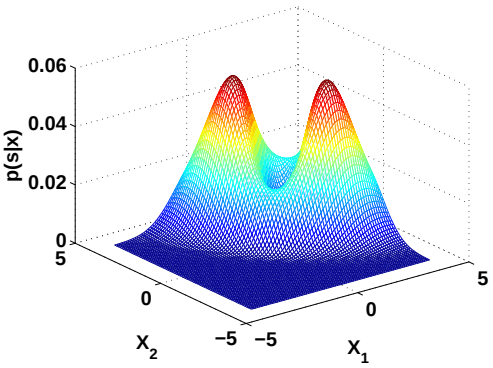


(c) UT approximation around $(3, 0)$.

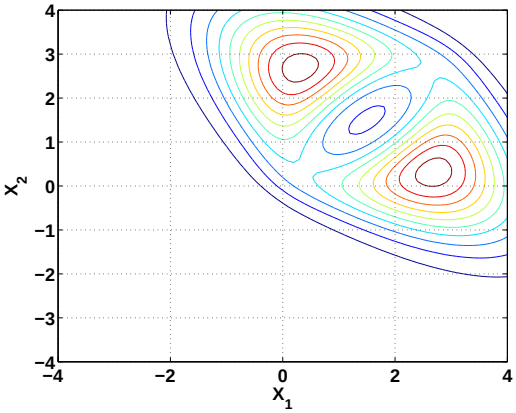


(d) UT approximation around $(3, 0)$.

Figure 8: "Likelihood" for two targets moving in one dimension; sum-of-powers SME.



(a) Analytical approximation around $(3, 0)$.



(b) Analytical approximation around $(3, 0)$.

Figure 9: "Likelihood" for two targets moving in one dimension; sum-of-products SME.

2.5 Analysis with Taylor Series Expansions

Several authors have presented Taylor series expansions about a random variable's mean as a method for comparing the accuracies of the EKF and UKF [16, 17, 41]. This section gives a brief introduction to this analysis tool and uses it to examine the filters' performances with the SME nonlinearities.

In this dissertation, the system has linear dynamics, but the SME formulation generates a nonlinear measurement equation. Under these conditions, the EKF uses the measurement covariance update

$$P_{\mathbf{y}_{k+1}\mathbf{y}_{k+1}} = \mathcal{J}_g P^-_{\mathbf{x}_{k+1}\mathbf{x}_{k+1}} \mathcal{J}_g^T + R_{k+1}, \tag{71}$$

where $g(\cdot)$ is the nonlinear SME function, $\mathcal{J}_g$ is the Jacobian matrix of $g(\cdot)$, and R is the measurement noise covariance matrix. The UKF uses

$$\mathbf{y}^-_{k+1} = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{Y}_{i,k+1|k}, \tag{72}$$

$$P_{\mathbf{y}_{k+1}\mathbf{y}_{k+1}} = \sum_{i=0}^{2n} W_i^{(c)} [\mathcal{Y}_{i,k+1|k} - \mathbf{y}^-_{k+1}][\mathcal{Y}_{i,k+1|k} - \mathbf{y}^-_{k+1}]^T, \tag{73}$$

where $\mathcal{X}$ is a set of sigma points in the state (Cartesian) space, $\mathcal{Y}$ is a set of sigma points in the observation (SME) space, and the W_i are the UKF weights defined in (32). The $\mathcal{J}_g P^-_{\mathbf{x}_{k+1}\mathbf{x}_{k+1}} \mathcal{J}_g^T$ term in (71) is a direct linearization of the covariance from the system space to the measurement space. We are interested in comparing these two methods, but (71) and (73) do not allow a direct comparison. Instead, we turn to Taylor series expansions to provide some insight.

Following the derivation in [41], we represent the initial variable, $\mathbf{x}$, as a zero-mean disturbance, $\delta\mathbf{x}$, about the mean, $\bar{\mathbf{x}}$. Then the Taylor series expansion of $g(\mathbf{x})$ becomes

$$g(\mathbf{x}) = g(\bar{\mathbf{x}} + \delta\mathbf{x}) = \sum_{n=0}^{\infty} \left[\frac{(\delta\mathbf{x} \cdot \nabla_{\mathbf{x}})^n g(\mathbf{x})}{n!} \right]_{\mathbf{x}=\bar{\mathbf{x}}}. \tag{74}$$

To simplify notation, we define

$$\mathbf{D}^n_{\delta\mathbf{x}} g \triangleq [(\delta\mathbf{x} \cdot \nabla_{\mathbf{x}})^n g(\mathbf{x})]_{\mathbf{x}=\bar{\mathbf{x}}} \tag{75}$$

so that (74) can be written as

$$\mathbf{y} = g(\mathbf{x}) = g(\bar{\mathbf{x}}) + \mathbf{D}_{\delta\mathbf{x}} g + \frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}} g + \frac{1}{3!}\mathbf{D}^3_{\delta\mathbf{x}} g + \ldots \tag{76}$$

2.5.1 Analyzing the Mean

From [41], the mean after the nonlinear transformation is

$$\bar{\mathbf{y}}_{\mathbf{T}} = \mathbb{E}[\mathbf{y}] = \mathbb{E}[g(\mathbf{x})], \tag{77}$$

$$= \mathbb{E}\left[g(\bar{\mathbf{x}}) + \mathbf{D}_{\delta\mathbf{x}}g + \frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}g + \frac{1}{3!}\mathbf{D}^3_{\delta\mathbf{x}}g + \dots\right], \tag{78}$$

where the subscript T indicates that the expression represents the true mean. Equation (78) can be simplified by assuming that $\mathbf{x}$ is a symmetrically distributed random variable, which causes all odd moments to equal zero. Two identities will also help simplify (78). First, $\mathbb{E}[\delta\mathbf{x}\delta\mathbf{x}^T] = P_{\mathbf{xx}}$. Second, $\mathbb{E}[\mathbf{D}^2_{\delta\mathbf{x}}g] = \frac{1}{2}[(\nabla^T P_{\mathbf{xx}}\nabla)g(\mathbf{x})]_{\mathbf{x}=\bar{\mathbf{x}}}$. This leads to

$$\bar{\mathbf{y}}_{\mathbf{T}} = g(\bar{\mathbf{x}}) + \frac{1}{2}[(\nabla^T P_{\mathbf{xx}}\nabla)g(\mathbf{x})]_{\mathbf{x}=\bar{\mathbf{x}}} + \mathbb{E}\left[\frac{1}{4!}\mathbf{D}^4_{\delta\mathbf{x}}g + \frac{1}{6!}\mathbf{D}^6_{\delta\mathbf{x}}g + \dots\right]. \tag{79}$$

In [41], the authors show that the unscented transform used in the UKF calculates the mean as

$$\bar{\mathbf{y}}_{\mathbf{UT}} = g(\bar{\mathbf{x}}) + \frac{1}{2}[(\nabla^T P_{\mathbf{xx}}\nabla)g(\mathbf{x})]_{\mathbf{x}=\bar{\mathbf{x}}} + \frac{1}{2(n+\lambda)}\sum_{i=1}^{2L}\left(\frac{1}{4!}\mathbf{D}^4_{\boldsymbol{\sigma}_i}g + \frac{1}{6!}\mathbf{D}^6_{\boldsymbol{\sigma}_i}g + \dots\right), \tag{80}$$

where $\boldsymbol{\sigma_i}$ denotes the i^{th} column of the matrix square root of $\sqrt{(n+\lambda)P_{\mathbf{xx}}}$, n is the state dimension, and λ is a scaling parameter. The linear approximation used in the EKF calculates the mean as

$$\bar{\mathbf{y}}_{\mathbf{LIN}} = g(\bar{\mathbf{x}}). \tag{81}$$

Comparing (79), (80), and (81), the unscented transform matches the true mean in the first and second-order terms, whereas the linearization only matches the true mean for the first term. Julier and Uhlmann show, in [16], that the sum of the errors in the higher order terms of $\bar{\mathbf{y}}_{\mathbf{UT}}$ are smaller than the error in assuming these terms are all zero, as the linear approximation does.

2.5.2 Mean Approximation for the Sum of Products SME

For the three-target sum-of-products case,

$$\nabla^T = \begin{bmatrix}\frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_3}\end{bmatrix}, \tag{82}$$

and describing $P_{\mathbf{xx}}$ as

$$P_{\mathbf{xx}} = \begin{bmatrix} \rho_{11} & \rho_{12} & \rho_{13} \\ \rho_{21} & \rho_{22} & \rho_{23} \\ \rho_{31} & \rho_{32} & \rho_{33} \end{bmatrix} \tag{83}$$

leads to

$$[(\nabla^T P_{\mathbf{xx}} \nabla) g_{prod}(\mathbf{x})]_{\mathbf{x}=\bar{\mathbf{x}}} = \begin{bmatrix} 0 \\ 2(\rho_{12} + \rho_{23} + \rho_{13}) \\ 2(\rho_{12}x_3 + \rho_{23}x_1 + \rho_{13}x_2) \end{bmatrix}. \tag{84}$$

Throughout this chapter, targets have been assumed to move independently, which makes the "true" $P_{\mathbf{xx}}$ a diagonal matrix. Since all the elements in the second-order term of (80), given in (84), depend only on the cross-covariances, the entire second-order term becomes zero if the filter estimates $P_{\mathbf{xx}}$ to be diagonal. Furthermore, the derivatives in the fourth and higher order terms of (80) reduce to zero for the three target case. This leaves $\bar{\mathbf{y}}_{\mathbf{UT}} = g(\bar{\mathbf{x}})$, which is exactly the same as $\bar{\mathbf{y}}_{\mathbf{LIN}}$. With the three-target sum-of-products SME and a diagonal $P_{\mathbf{xx}}$, the UKF and EKF employ the same approximation for the measurement noise mean after the SME transformation.

2.5.3 Mean Approximation for the Sum of Powers SME

Following the derivation from the previous section, the second-order term in the Taylor expansion for the sum-of-powers SME is

$$\left[(\nabla^T P_{\mathbf{xx}} \nabla) g_{pow}(\mathbf{x})\right]_{\mathbf{x}=\bar{\mathbf{x}}} = \begin{bmatrix} 0 \\ 2(\rho_{11} + \rho_{22} + \rho_{33}) \\ 6(\rho_{11}x_1 + \rho_{22}x_2 + \rho_{33}x_3) \end{bmatrix}. \tag{85}$$

The second-order term, (85), depends only on the variance terms of $P_{\mathbf{xx}}$. Consequently, the second-order term is not zero, although the derivatives in the fourth and higher order terms reduce to zero for a three-target case. Thus, with one more non-zero term than the EKF, the UKF will provide a better estimate of the mean than the linearization used by the EKF.

2.5.4 Analyzing the Covariance

The Taylor series for the true covariance [41] is given by

$$\begin{aligned} P_{\mathbf{yy}} = \mathcal{J}_{\mathbf{x}} P_{\mathbf{xx}} \mathcal{J}_{\mathbf{x}}^T & \\ -\frac{1}{4}\Big\{ \big[(\nabla^T P_{\mathbf{xx}} \nabla) g(\mathbf{x})\big] \big[(\nabla^T P_{\mathbf{xx}} \nabla) g(\mathbf{x})\big]^T \Big\}_{\mathbf{x}=\bar{\mathbf{x}}} & \\ + \mathbb{E} \underbrace{\left[\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \frac{1}{i!j!} \mathbf{D}^i_{\delta\mathbf{x}} g \ (\mathbf{D}^j_{\delta\mathbf{x}} g)^T \right]}_{i \neq j = 1} & \\ - \underbrace{\left[\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \frac{1}{(2i)!(2j)!} \mathbb{E}\big[\mathbf{D}^{2i}_{\delta\mathbf{x}} g\big] \mathbb{E}\big[\mathbf{D}^{2j}_{\delta\mathbf{x}} g\big]^T \right]}_{i \neq j = 1}, & \end{aligned} \tag{86}$$

where $\mathcal{J}_{\mathbf{x}}$ is the Jacobian matrix of $g(\mathbf{x})$ evaluated at $\mathbf{x} = \bar{\mathbf{x}}$.

Using a similar approach as with the mean, the authors of [16] showed that the unscented transform calculates the transformed covariance as

$$\begin{aligned} (P_{\mathbf{yy}})_{UT} = \mathcal{J}_{\mathbf{x}} P_{\mathbf{xx}} \mathcal{J}_{\mathbf{x}}^T & \\ -\frac{1}{4}\Big\{ \big[(\nabla^T P_{\mathbf{xx}} \nabla) g(\mathbf{x})\big] \big[(\nabla^T P_{\mathbf{xx}} \nabla) g(\mathbf{x})\big]^T \Big\}_{\mathbf{x}=\bar{\mathbf{x}}} & \\ + \frac{1}{2(n+\lambda)} \sum_{k=1}^{2L} \underbrace{\left[\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \frac{1}{i!j!} \mathbf{D}^i_{\sigma_{\mathbf{k}}} g \ (\mathbf{D}^j_{\sigma_{\mathbf{k}}} g)^T \right]}_{i \neq j = 1} & \\ - \underbrace{\left[\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \frac{1}{(2i)!(2j)!4(n+\lambda)^2} \sum_{k=1}^{2L} \sum_{m=1}^{2L} \big[\mathbf{D}^{2i}_{\sigma_{\mathbf{k}}} g\big] \big(\mathbf{D}^{2j}_{\sigma_{\mathbf{m}}} g\big)^T \right]}_{i \neq j = 1}. & \end{aligned} \tag{87}$$

Comparing (87) and (87), the unscented transform correctly calculates the first two terms of the covariance. The linearization, on the other hand, calculates the transformed covariance as

$$(P_{\mathbf{yy}})_{LIN} = \mathcal{J}_{\mathbf{x}} P_{\mathbf{xx}} \mathcal{J}_{\mathbf{x}}^T, \tag{88}$$

which is only accurate for the first term. Again, the authors of [16] show that the error for unscented transform's approximation is less than the assumption that all higher order terms are zero.

2.5.5 Covariance Approximation for the Sum of Products SME

Section 2.5.2 showed that the second term of (87) is zero for the sum-of-products when the initial covariance matrix is diagonal. The third and fourth terms are also zero for the sum-of-products when the initial covariance matrix is diagonal. This is true because any element of these terms that is non-zero after taking all the appropriate derivatives is multiplied by a cross-covariance that is zero. The unscented transform's mean and covariance produce an approximation similar to that of the linearization used in the EKF for the sum-of-products SME. Unfortunately, a direct comparison of the EKF and UKF is not practical because the UKF is incompatible with the sum of products.

2.5.6 Covariance Approximation for the Sum of Powers SME

Section 2.5.3 showed that the second term of (87) depends on the variance terms of the initial covariance matrix and is nonzero. Thus, the unscented transform generates a more accurate approximation of the measurement probability density than the linearization used in the EKF. The effect of this improved approximation can be seen in the results. The UKF has significantly better performance than the EKF when paired with the sum-of-powers SME.

2.5.7 Other Observations from Taylor Series Analysis

2.5.7.1 Data Dependence

Equations (84) and (85) reveal one drawback to the SME method. Since the data appears in the third row of the expression, the covariance estimate is a function of the data. This is equivalent to saying that the covariance depends upon the location of the mean. While this problem can be alleviated by using a centered coordinate system, ideally the covariance estimate would be independent of the raw data's Cartesian origin.

2.5.7.2 Increasing the Number of Targets

The multiple target tracking problem increases in complexity as the number of targets increases. In the SME approach, this is manifested by an increase in the number of equations needed, the magnitude of the numbers used in these equations, and the difficulty posed by increasing the number of variables maintained by the filter. On top of these problems,

our Taylor series analysis demonstrates that increasing the number of targets reduces the accuracy of the EKF and UKF's approximations. The EKF is only accurate for the first term of the Taylor series and the UKF is only accurate for the first two terms. The derivatives in the higher-order Taylor-series terms generally cause those terms to be zero. Unfortunately, as the number of targets increases, the number of nonzero terms increases. Since the number of nonzero terms unaccounted for by the EKF or UKF relates to the approximation error of these filters, increasing the number of targets increases the error in the filters' approximations. This is a limitation of using the EKF or UKF for SME-based tracking rather than a limitation of the SME approach itself.

2.6 Conclusions

Several interesting phenomena were observed in this chapter. Kamen's early studies found the sum-of-products form of the SME to work better than the sum of powers with the EKF; hence, later work by him and his colleagues focused on the sum-of-products form. We have found that an UKF implementation of the sum-of-powers – the form originally abandoned by Kamen - actually performs better than the EKF implementation of either form. Also, we have discovered that the sum-of-products nonlinearity is inherently incompatible with the UKF, inadvertently uncovering an aspect of UKFs that seems to be frequently overlooked in the literature. These results suggest that the performance of the SME approach is dependent on the pairing of a specific SME implementation and nonlinear filter rather than dependent on either individually.

CHAPTER III

A NEW IMPLEMENTATION OF TWO-DIMENSIONAL SME TRACKING

Inspired by the performance of the unscented Kalman filter (UKF) paired with the sum-of-powers SME seen in Chapter 2, we decided to explore two-dimensional tracking with the SME. An efficient implementation of a two-dimensional sum-of-products SME-based tracker already existed, but no such two-dimensional implementation using the sum-of-powers existed. This chapter presents our two-dimensional sum-of-powers SME along with results from filtering with an EKF, an UKF, and a particle filter.

3.1 Background

Extending the SME approach to multidimensional tracking may, at first, seem straightforward. It is easy to imagine taking each dimension of the Cartesian coordinate system and applying SME functions separately. For instance, in the case of two-dimensional tracking of three targets using the SME approach, the equations might be

$$\tilde{z}_{prod} = \begin{pmatrix} x_1 + x_2 + x_3 \\ x_1x_2 + x_1x_3 + x_2x_3 \\ x_1x_2x_3 \\ y_1 + y_2 + y_3 \\ y_1y_2 + y_1y_3 + y_2y_3 \\ y_1y_2y_3 \end{pmatrix} \quad \text{or} \quad \tilde{z}_{pow} = \begin{pmatrix} x_1 + x_2 + x_3 \\ x_1^2 + x_2^2 + x_3^2 \\ x_1^3 + x_2^3 + x_3^3 \\ y_1 + y_2 + y_3 \\ y_1^2 + y_2^2 + y_3^2 \\ y_1^3 + y_2^3 + y_3^3 \end{pmatrix}. \tag{89}$$

Indeed, this approach was the first tried [35]. Unfortunately, it has a significant drawback. SME-based filters using this kind of formulation exhibit "coordinate-switching." For example, the filter might pair the x-coordinate of target 1 with the y-coordinate of target 2 and vice-versa [31]. This results in "ghost targets," where the filter tracks a target that does not

exist. This is a significantly worse failure mode than the typical track-switching problem that all multiple target tracking algorithms may experience.

A clever solution for the coordinate switching problem in two dimensions was presented in [24] and [31]. The idea is to encode one coordinate of a Cartesian observation in the real part of a complex number and the other coordinate as the imaginary part. A set of SME functions can then be applied to the complex observations. This previously published method is based on the sum-of-products SME. Suppose that the complex number p_i represents the i^{th} two-dimensional observation, with the x-coordinate as the real part and the y-coordinate as the imaginary part, such that $p_i = x_i + jy_i$. For the three-target case, the sum-of-products two-dimensional SME equations are

$$\tilde{z}_{cprod} = \begin{pmatrix} \mathcal{R}\{p_1 + p_2 + p_3\} \\ \mathcal{I}\{p_1 + p_2 + p_3\} \\ \mathcal{R}\{p_1p_2 + p_1p_3 + p_2p_3\} \\ \mathcal{I}\{p_1p_2 + p_1p_3 + p_2p_3\} \\ \mathcal{R}\{p_1p_2p_3\} \\ \mathcal{I}\{p_1p_2p_3\} \end{pmatrix}, \tag{90}$$

which expands to

$$\tilde{z}_{cprod} = \begin{pmatrix} x_1 + x_2 + x_3 \\ y_1 + y_2 + y_3 \\ x_1x_2 + x_1x_3 + x_2x_3 - y_1y_2 - y_1y_3 - y_2y_3 \\ x_1y_2 + x_2y_1 + x_1y_3 + x_3y_1 + x_2y_3 + x_3y_2 \\ x_1x_2x_3 - (y_1y_2x_3 + y_1y_3x_2 + y_2y_3x_1) \\ x_1x_2y_3 + x_1x_3y_2 + x_2x_3y_1 - y_1y_2y_3 \end{pmatrix}, \tag{91}$$

where the subscript *cprod* indicates a sum-of-products over the complex numbers. Complete details for implementing a SME tracking system with this SME can be found in Lee's thesis [24].

Ref. [31] presents another solution for performing N-dimensional SME tracking without

the coordinate-switching/ghost-target problem. Their approach encodes different coordinates of a measurement as coefficients of a polynomial. The downside to their technique is that it requires a larger number of SME equations than other approaches, and consequently demands higher complexity and computational resources. In the two-dimensional scenario, their approach requires $(N^2 + 3N)/2$ equations, where N is the number of targets. The complex number approach explored in this chapter only requires $2N$ equations.

3.2 A New Two-Dimensional SME

Motivated by the observation from Chapter 2 that, in one dimension, the sum-of-powers SME offered better performance with the UKF than could be achieved with the sum-of-products, we developed a sum-of-powers two-dimensional SME. Following the same approach with complex numbers as given in the previous section, our new SME for two targets is given by

$$\tilde{z}_{cpow} = \begin{pmatrix} x_1 + x_2 \\ y_1 + y_2 \\ x_1^2 + x_2^2 - y_1^2 - y_2^2 \\ 2x_1y_1 + 2x_2y_2 \end{pmatrix}, \tag{92}$$

and for three targets, it is given by

$$\tilde{z}_{cpow} = \begin{pmatrix} x_1 + x_2 + x_3 \\ y_1 + y_2 + y_3 \\ x_1^2 + x_2^2 + x_3^2 - y_1^2 - y_2^2 - y_3^2 \\ 2x_1y_1 + 2x_2y_2 + 2x_3y_3 \\ x_1^3 + x_2^3 + x_3^3 - 3x_1y_1^2 - 3x_2y_2^2 - 3x_3y_3^2 - 2\sigma^2(x_1 + x_2 + x_3) \\ -y_1^3 - y_2^3 - y_3^3 + 3y_1x_1^2 + 3y_2x_2^2 + 3y_3x_3^2 - 2\sigma^2(y_1 + y_2 + y_3) \end{pmatrix}, \tag{93}$$

where σ^2 is the observation noise variance.[1] The terms in the fifth and sixth line of (93) that are scaled by σ^2 ensure that this set of SME equations generates additive zero-mean observation noise (see Appendix B for more explanation).

[1] To make the equations palatable, each measurement was assumed to have independent observation noise with the same variance such that $\sigma^2 = \sigma^2_{x_1} = \sigma^2_{x_2} = ... = \sigma^2_{x_N} = \sigma^2_{y_1} = \sigma^2_{y_2} = ... = \sigma^2_{y_N}$.

3.2.1 Existence of the Inverse SME

One of the key requirements in the choice of SME functions is that the functions preserve all the original information in the data [19]. Suppose N SME pseudomeasurements, s_i, are the result of applying the SME functions, $g_i(\cdot)$, to the real measurements, m_i, such that

$$s_i = g_i(m_1, \cdots, m_N) \quad \text{for } i = 1, 2, \ldots, N. \tag{94}$$

To ensure that the SME transformation has preserved all the information in the original m_i measurements, another set of equations $g_i^{-1}(\cdot)$ must exist such that

$$m_i = g_i^{-1}(s_1, \cdots, s_N) \quad \text{for } i = 1, 2, \ldots, N. \tag{95}$$

As long as all the m_i can be recovered, up to a permutation, then no information has been lost. Since the SME approach is associationless, the order of the original inputs was never important.

One method of proving the existence of all the $g_i^{-1}(\cdot)$ would be to find them. For all but the most trivial cases, this is quite difficult. Fortunately, there is an alternative. By the Implicit Function Theorem [6,19], we know that if the determinant of the Jacobian, $\mathcal{J}$, with respect to the original inputs, m, of the symmetric measurement equations $g(\cdot)$ is nonzero, then an inverse of the SME equations exists at m. One way to intuitively understand the Implicit Function Theorem is to relate it to the case of linear equations. From linear algebra, we know that if a set of linear equations are written in matrix form, then the equations are linearly independent if their coefficient matrix is full rank (equivalently, the determinant of the matrix is nonzero).[2] Equivalently, if the determinant of the Jacobian matrix for a set of nonlinear equations is nonzero, then the nonlinear equations are, in some sense, "independent."

For the two-dimensional case, the m_i measurements are denoted as x_i or y_i for $i = 1, \cdots, N/2$. The Jacobian matrix, $\mathcal{J}$, for the two-dimensional, two-target sum-of-powers

[2]The familiar rule for linear equations is, as one would expect, a special case of the more general Implicit Function Theorem.

SME is given by

$$\mathcal{J}_{2T} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 2x_1 & -2y_1 & 2x_2 & -2y_2 \\ 2y_1 & 2x_1 & 2y_2 & 2x_2 \end{pmatrix}, \tag{96}$$

and the determinant of $\mathcal{J}_{2T}$ is

$$det(\mathcal{J}_{2T}) = -4\left[(x_2 - x_1)^2 + (y_2 - y_1)^2\right]. \tag{97}$$

For the three target case, the Jacobian is given by

$$\mathcal{J}_{3T} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 2x_1 & -2y_1 & 2x_2 & -2y_2 & 2x_3 & -2y_3 \\ 2y_1 & 2x_1 & 2y_2 & 2x_2 & 2y_3 & 2x_3 \\ 3(x_1^2 - y_1^2) - 2\sigma^2 & -6x_1y_1 & 3(x_2^2 - y_2^2) - 2\sigma^2 & -6x_2y_2 & 3(x_3^2 - y_3^2) - 2\sigma^2 & -6x_3y_3 \\ 6x_1y_1 & 3(x_1^2 - y_1^2) - 2\sigma^2 & 6x_2y_2 & 3(x_2^2 - y_2^2) - 2\sigma^2 & 6x_3y_3 & 3(x_3^2 - y_3^2) - 2\sigma^2 \end{pmatrix}, \tag{98}$$

and the determinant of $\mathcal{J}_{3T}$ is

$$det(\mathcal{J}_{3T}) =$$

$$-36\left([(x_2 - x_1)^2 + (y_2 - y_1)^2][(x_3 - x_2)^2 + (y_3 - y_2)^2][(x_3 - x_1)^2 + (y_3 - y_1)^2]\right). \tag{99}$$

Clearly, the only time (97) or (99) would be zero is when two targets are located at the same place. In such cases, we would expect the number of "independent" equations to be reduced. When the targets are crossing ($m_i = m_j$), we can show that there are still the necessary number of "independent" equations to recover the two unique positions of the three targets, but refrain from doing so here as it would be repetitive.

Computing the determinant for the two-dimensional sum-of-products provides an interesting result. For the two-dimensional two target sum-of-products case, the determinant of

the Jacobian is

$$det(\mathcal{J}_{2T}) = -\left[(x_2 - x_1)^2 + (y_2 - y_1)^2\right], \tag{100}$$

and for the three target case it is

$$det(\mathcal{J}_{3T}) =$$

$$-\left([(x_2 - x_1)^2 + (y_2 - y_1)^2][(x_3 - x_2)^2 + (y_3 - y_2)^2][(x_3 - x_1)^2 + (y_3 - y_1)^2]\right). \tag{101}$$

Comparing (97) to (100) and (99) to (101) shows that the determinants for the sum-of-products and the sum-of-powers are identical up to a constant scaling factor.

3.3 Nonlinear Filters for the Two-Dimensional SME

To implement the SME tracking filter using an EKF, the Jacobian matrix of the SME and a Gaussian approximation for the observation noise need to be found. These computations are quite tedious and the resulting expressions quite lengthy, especially the Gaussian noise approximation. The formulas for the sum-of-products can be found in [24], and the formulas we derived for the sum-of-powers case are in Appendix B. Examining the equations for the Gaussian approximations for both the sum-of-products and the sum-of-powers makes the appeal of the UKF clear, since it does not require these cumbersome calculations as discussed in Chapter 2.

The particle filter is a straightforward extension of the particle filter in Chapter 2. Each particle contains an entry for position and velocity in both dimensions for each target. This design leads to particles with a large dimension, which adversely affects particle filter performance. However, since we are avoiding explicit data association, there is no readily apparent alternative. Also, as in the one-dimensional case, the exact likelihood of the SME pseudomeasurement given the current state $p(\mathbf{s}|\mathbf{x})$ is difficult to find. Instead, as in Chapter 2, we use two methods for generating Gaussian approximations of the likelihood: the analytical approximation from the EKF and the unscented transform from the UKF.

3.4 Analysis with Taylor Series Expansions

In Section 2.5, we presented Taylor series expansions as a method for comparing the accuracies of the EKF and UKF. This section presents this analysis for the two-dimensional

SME implementations discussed in this chapter. Recall that the Taylor series expansion of the true mean estimate is

$$\bar{\mathbf{y}}_{\mathbf{T}} = g(\bar{\mathbf{x}}) + \frac{1}{2}[(\nabla_{\mathbf{x}}^T P_{\mathbf{xx}} \nabla_{\mathbf{x}}) g(\mathbf{x})]_{\mathbf{x}=\bar{\mathbf{x}}} + \mathbb{E}\left[\frac{1}{4!}\mathbf{D}_{\delta\mathbf{x}}^4 g + \frac{1}{6!}\mathbf{D}_{\delta\mathbf{x}}^6 g + \dots\right]. \tag{102}$$

The unscented transform used in the UKF calculates the mean as

$$\bar{\mathbf{y}}_{\mathbf{UT}} = g(\bar{\mathbf{x}}) + \frac{1}{2}[(\nabla_{\mathbf{x}}^T P_{\mathbf{xx}} \nabla_{\mathbf{x}}) g(\mathbf{x})]_{\mathbf{x}=\bar{\mathbf{x}}} + \frac{1}{2(n+\lambda)} \sum_{i=1}^{2n} \left(\frac{1}{4!}\mathbf{D}_{\boldsymbol{\sigma}_i}^4 g + \frac{1}{6!}\mathbf{D}_{\boldsymbol{\sigma}_i}^6 g + \dots\right), \tag{103}$$

where $\boldsymbol{\sigma_i}$ denotes the i^{th} column of the matrix square root $\sqrt{(n+\lambda)P_{\mathbf{xx}}}$, n is the state dimension, and λ is the composite UKF scaling parameter. The linear approximation used in the EKF calculates the mean as

$$\bar{\mathbf{y}}_{\mathbf{LIN}} = g(\bar{\mathbf{x}}). \tag{104}$$

Notice that both the unscented transform and the linearization are accurate for the first order term, $g(\bar{\mathbf{x}})$.

For the two-dimensional three-target SME, the state variable, $\mathbf{x}$, is

$$\mathbf{x}^T = \begin{bmatrix} x_1 & y_1 & x_2 & y_2 & x_3 & y_3 \end{bmatrix}. \tag{105}$$

To simplify notation, however, the elements of $\mathbf{x}$ will be denoted as $x_1, x_2, \dots, x_6$. For the three target case, ∇ is

$$\nabla^T = \begin{bmatrix} \dfrac{\partial}{\partial x_1} & \dfrac{\partial}{\partial x_2} & \dfrac{\partial}{\partial x_3} & \dfrac{\partial}{\partial x_4} & \dfrac{\partial}{\partial x_5} & \dfrac{\partial}{\partial x_6} \end{bmatrix}, \tag{106}$$

and $P_{\mathbf{xx}}$ is

$$P_{\mathbf{xx}} = \begin{bmatrix} \rho_{11} & \rho_{12} & \rho_{13} & \rho_{14} & \rho_{15} & \rho_{16} \\ \rho_{21} & \rho_{22} & \rho_{23} & \rho_{24} & \rho_{25} & \rho_{26} \\ \rho_{31} & \rho_{32} & \rho_{33} & \rho_{34} & \rho_{35} & \rho_{36} \\ \rho_{41} & \rho_{42} & \rho_{43} & \rho_{44} & \rho_{45} & \rho_{46} \\ \rho_{51} & \rho_{52} & \rho_{53} & \rho_{54} & \rho_{55} & \rho_{56} \\ \rho_{61} & \rho_{62} & \rho_{63} & \rho_{64} & \rho_{65} & \rho_{66} \end{bmatrix}. \tag{107}$$

3.4.1 Mean Approximation for the Two-Dimensional Sum-of-Powers SME

Since both the EKF and the UKF correctly model the first term of the Taylor series for the mean, we begin our analysis with the second-order term. The second-order term for the sum-of-powers SME is

$$\left[(\nabla^T P_{\mathbf{xx}} \nabla) g_{pow}(\mathbf{x})\right]_{\mathbf{x}=\bar{\mathbf{x}}} =$$

$$\begin{bmatrix} 0 \\ 0 \\ 2(\rho_{11} - \rho_{22} + \rho_{33} - \rho_{44} + \rho_{55} - \rho_{66}) \\ 4(\rho_{12}\rho_{21} + \rho_{34}\rho_{43} + \rho_{56}\rho_{65}) \\ 6(\rho_{11}x_1 + \rho_{33}x_3 + \rho_{55}x_5 - \rho_{21}x_2 - \rho_{22}x_1 - \rho_{43}x_4 - \rho_{44}x_3 - \rho_{65}x_6 - \rho_{66}x_5) \\ 6(\rho_{11}x_2 + \rho_{33}x_4 + \rho_{55}x_6 + \rho_{21}x_1 + \rho_{43}x_3 + \rho_{65}x_5 - \rho_{22}x_2 - \rho_{44}x_4 - \rho_{66}x_6) \end{bmatrix}. \quad (108)$$

Throughout this thesis, the true targets move independently of each other and with independent motion in the horizontal and vertical dimensions. When the filters believe this is the case, $P_{\mathbf{xx}}$ becomes a diagonal or nearly-diagonal matrix, and $\rho_{ij} \approx 0$ for $i \neq j$. Because the second-order term, (108), contains the diagonal terms of $P_{\mathbf{xx}}$, it will not reduce to zero, although the derivatives in the fourth-order and higher-order terms reduce these terms to zero for a three-target case. Thus, with one more non-zero term than the EKF, the UKF should provide a better estimate of the mean than the linearization used by the EKF. We will see, however, that for different reasons this assertion does not hold.

3.4.2 Mean Approximation for the Two-Dimensional Sum-of-Products SME

The second-order term for the sum-of-products SME is

$$\left[(\nabla^T P_{\mathbf{xx}} \nabla) g_{prod}(\mathbf{x})\right]_{\mathbf{x}=\bar{\mathbf{x}}} =$$

$$\begin{bmatrix} 0 \\ 0 \\ 2(\rho_{31}+\rho_{51}+\rho_{35}-\rho_{42}-\rho_{62}-\rho_{64}) \\ 2(\rho_{41}+\rho_{32}+\rho_{61}+\rho_{52}+\rho_{63}+\rho_{54}) \\ 2((\rho_{31}-\rho_{42})x_5+(\rho_{51}-\rho_{62})x_3+(\rho_{53}-\rho_{64})x_1-(\rho_{41}+\rho_{32})x_6-(\rho_{61}+\rho_{52})x_4-(\rho_{54}+\rho_{63})x_2) \\ 2((\rho_{31}-\rho_{42})x_6+(\rho_{51}-\rho_{62})x_4+(\rho_{53}-\rho_{64})x_2+(\rho_{41}+\rho_{32})x_5+(\rho_{61}+\rho_{52})x_3+(\rho_{54}+\rho_{63})x_1) \end{bmatrix}. \quad (109)$$

Equation (109) was simplified by recognizing that $\rho_{ij} = \rho_{ji}$ for a covariance matrix. Recalling the assumption that targets move independently of each other and with independent motion ($\rho_{ij} \approx 0$ for $i \neq j$), all the elements of the second-order term, (109), become zero, at least if the filters accurately model this behavior. Furthermore, the derivatives in the fourth and higher-order terms of (103) reduce to zero for the three-target case. This leaves $\bar{\mathbf{y}}_{\mathbf{UT}} = g(\bar{\mathbf{x}})$, which is exactly the same as $\bar{\mathbf{y}}_{\mathbf{LIN}}$. With the three-target sum-of-products SME and a diagonal $P_{\mathbf{xx}}$, the UKF and EKF employ the same approximation for the measurement noise mean after the SME transformation.

Examining this analysis from another viewpoint, it suggests that the sum-of-products SME is relatively easy to approximate, at least in a Taylor series sense, since the second-order Taylor series term for the sum-of-products reduces to zero. On the other hand, the sum-of-powers nonlinearity has a nonzero second-order term, and consequently is more difficult to model, at least by truncated Taylor series, than the sum-of-products.

3.5 Results

3.5.1 Simulation Setup

The simulations consisted of three targets moving in two dimensions. Each target moved independently with nearly constant velocity and a known process noise variance. Initial positions and velocities were chosen so that the targets were likely to cross paths.[3] The observation data was generated by adding Gaussian noise with zero mean and a known variance. Observations were assumed to come from "magical" sensors providing Cartesian

[3]The process noise was kept low so that targets cross at nearly the same time, although they will rarely pass through the exact same place at the same time.

coordinates with uncorrelated errors in both dimensions.[4] Software developed by Wan and van der Merwe provided the basis for these simulations [42].

Extensive simulations were run for two scenarios. An example of the first scenario is shown in Figure 10. Two targets start on the shown vertical axis and the third on the shown horizontal axis. Two of the three targets pass through similar points at the same time. This scenario may be considered relatively easy because the crossing targets are moving nearly perpendicularly. The targets are all moving at around 200 m/s, and positions on the axes are given in meters. Figure 11 shows an example of a more challenging tracking scenario

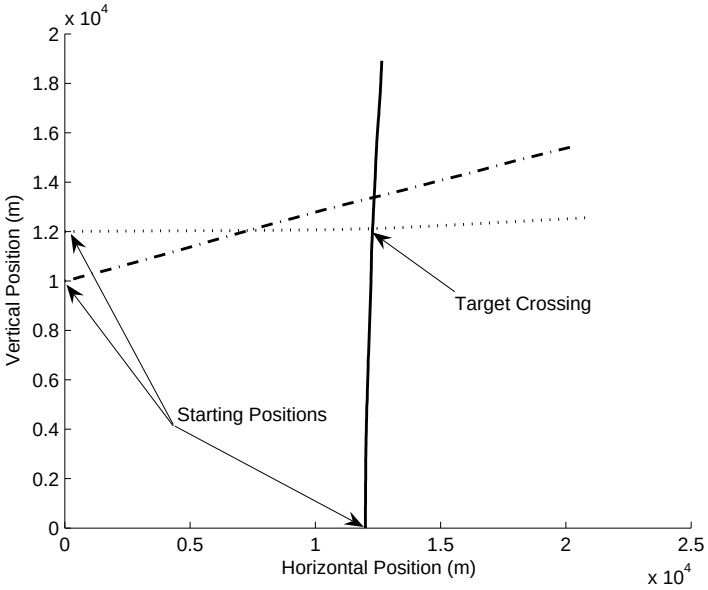


Figure 10: Scenario 1.

where the three targets cross paths at similar times and are moving in similar directions. All tracks were initialized with the first position estimate being the fifth observation. The first four observations were used to generate initial velocity estimates. For the particle filters, the velocity components of the initial particles were distributed with a Gaussian distribution around these estimated velocities. Also, a global nearest-neighbor (GNN) algorithm and an associated Kalman filter implementation were included in all simulations for comparison. The GNN approach represents a simple solution to the multiple target tracking problem.

[4]The uncorrelated error assumption greatly simplifies the analysis in Appendix B.

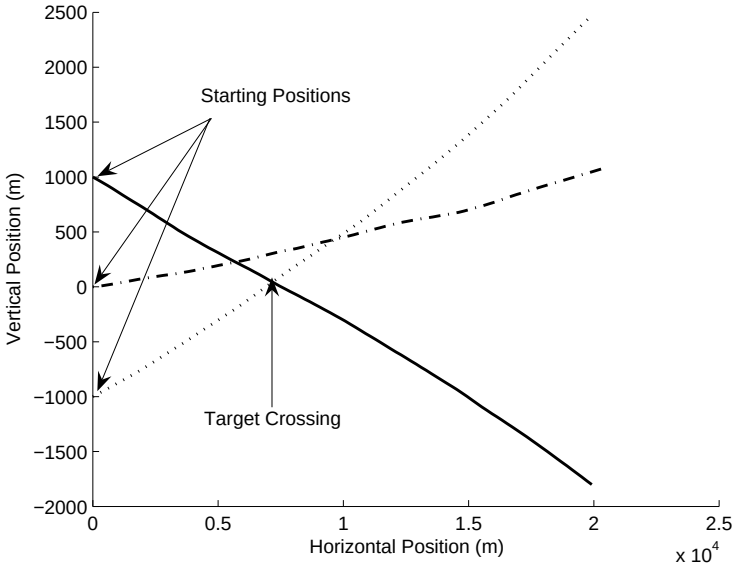


Figure 11: Scenario 2.

Since the simulation scenario does not include false alarms and missed detections, each observation is used to update exactly one track. The GNN algorithm associates observations with tracks such that the sum of the distances from the observations to the predicted positions is minimized [5]. This approach may also be thought of as a 2-D assignment algorithm. Separate standard Kalman filters are run for each track. The associated KF, on the other hand, has knowledge of the correct target associations and runs a standard Kalman filter for each track.

3.5.2 Data

For this set of results, the observation noise variance was set to 1×10^4 m^2. This high value was chosen to induce the tracking filters to switch tracks at least a small percentage of the time. The process noise variance was set to 1 m^2/s^4, and 1000 Monte Carlo trials were performed.

Tables 5 and 6 show the percentage of the total simulation runs in which each SME-filter pairing maintained target associations. Tables 7 and 8 give the average RMSE. Only simulation runs in which the filter maintained correct target associations are included in this calculation. Tables 9 and 10 give the "set estimation" RMSE. The set estimation RMSE

Table 5: Percentage of Trials with Correctly Maintained Associations - Scenario 1

	2D Sum of Products	2D Sum of Powers
Extended Kalman	99	100
Unscented Kalman	97	94
PF - Analytical	83	80
PF - Unscented	81	79
Global NN	99	
Associated KF	100	

Table 6: Percentage of Trials with Correctly Maintained Associations - Scenario 2

	2D Sum of Products	2D Sum of Powers
Extended Kalman	89	86
Unscented Kalman	80	71
PF - Analytical	26	26
PF - Unscented	29	25
Global NN	86	
Associated KF	100	

is calculated by finding the track-estimate/track-truth association with the smallest MSE at each time step and then averaging over all time steps. Consequently, target associations are not considered.

For Scenario 1 (the easier scenario), performance seems to be pretty even between the two SME forms. However, in the more challenging scenario, the sum-of-products has a significant advantage over the sum-of-powers. The UKF is also generally worse than the EKF. These trends run against our findings for the one-dimensional SME (see Chapter, 2 where the UKF paired with the sum-of-powers exhibited the best performance [26,27]. The most noticeable trend is the surprisingly poor performance of the particle filters.

3.6 Difficulties with Nonlinear Filters

3.6.1 The UKF

The proponents of the UKF suggest that it should always outperform the EKF. Indeed, even our own analysis suggests the UKF should outperform the EKF in our application (3.4.1). Nonetheless, Chapter 2.1.3 showed that the one-dimensional sum-of-products SME

Table 7: Associated Root Mean Squared Error - Scenario 1

	2D Sum of Products	2D Sum of Powers
Extended Kalman	58.9	59.9
Unscented Kalman	60.3	62.9
PF - Analytical	1435	1466
PF - Unscented	1489	1484
Global NN	55.2	
Associated KF	55.3	

Table 8: Associated Root Mean Squared Error - Scenario 2

	2D Sum of Products	2D Sum of Powers
Extended Kalman	63.4	102.8
Unscented Kalman	71.8	138.1
PF - Analytical	1760	1778
PF - Unscented	1739	2000
Global NN	63.1	
Associated KF	55.8	

is incompatible with the UKF. This stems from the limitation that the UKF only guarantees a positive semidefinite estimate of the covariance matrix [44], while a positive definite estimate is needed for the filter to work properly. In most situations, the UKF's estimate is positive definite and the filter runs well. However, in the sum-of-products one-dimensional case, the estimate becomes singular when targets are located close to each other.

It turns out that the same problem arises in the two-dimensional case, but now with both the two-dimensional sum-of-products and the two-dimensional sum-of-powers. When target estimates are crossing, the estimated covariance is no longer numerically positive definite. In terms of the UKF's sigma points, this condition is recognizable as a loss in the dimensionality of the sigma points. For example, in the two-target, two-dimensional case, the sigma points can be thought of as occupying four dimensions.[5] When targets cross paths, the dimension is reduced to two or three. To make visualization of this case possible, we briefly ignore one dimension of the sigma points. Figure 12(a) shows the sigma points

[5]The sigma points actually occupy a larger space since the state has been augmented to include observation and process noise, but that is not relevant to our discussion here.

Table 9: Set Estimation Root Mean Squared Error - Scenario 1

	2D Sum of Products	2D Sum of Powers
Extended Kalman	59.3	59.8
Unscented Kalman	60.5	63.8
PF - Analytical	1357	1408
PF - Unscented	1412	1417
Global NN	56.3	
Associated KF	55.3	

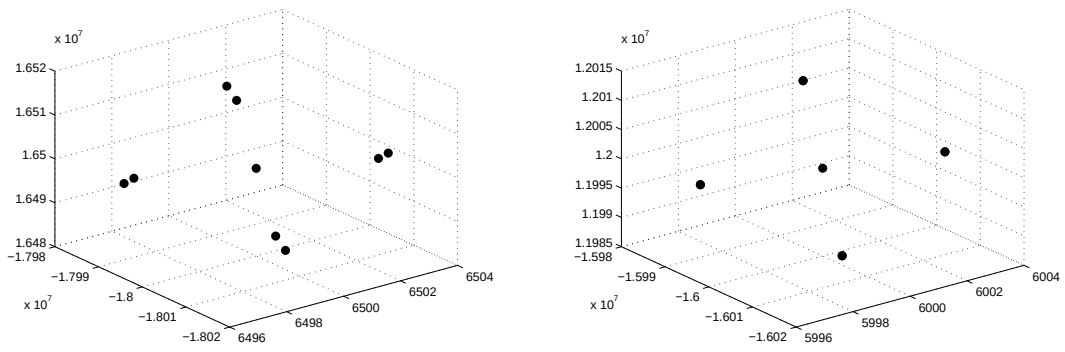
Table 10: Set Estimation Root Mean Squared Error - Scenario 2

	2D Sum of Products	2D Sum of Powers
Extended Kalman	65.0	109.4
Unscented Kalman	71.6	116.9
PF - Analytical	1464	1417
PF - Unscented	1477	1501
Global NN	64.4	
Associated KF	55.7	

when the two target estimates are closely spaced, and Figure 12(b) shows the sigma points when the two target estimates are crossing. When the target estimates are crossing, we see that the sigma points no longer occupy a full three-dimensional space, but instead all fall in a single plane. This is analogous to Figure 3, where the one-dimensional sum-of-products caused the sigma points to lie on a line rather than to define a plane when target estimates are crossing. Examining the equations for the two-dimensional sum-of-powers SME, it is not surprising that the UKF has trouble. The terms in (92) and (93) resemble those of the one-dimensional sum-of-products SME given in (21) that plagued the UKF earlier.

3.6.2 The Particle Filter

Difficulties with the particle filter in one-dimensional SME-based tracking were discussed in detail in Section 2.4. The trends discussed there are even more dramatic for the two-dimensional SME cases, as would be expected from the increase in dimensionality. Using the prior as the particle filter proposal density places few particles near observations, and using more particles to overcome this problem requires an impractical quantity of particles.



(a) When target estimates are closely spaced.

(b) When target estimates are crossing.

Figure 12: UKF sigma points for two targets in two dimensions.

Another option would be to generate a better proposal distribution that takes into account the prior and the observation. However, the particle filters, as implemented, have another problem: the Gaussian approximations used in the likelihood density do not place the highest likelihood at the true locations of the targets. To study this problem, we re-ran the Scenario 1 simulation with a lower observation noise (10 m^2) and the same process noise (1 $\mathrm{m}^2/\mathrm{s}^2$) to mitigate the effects of the first problem. Table 11 gives the resulting set error. With this setup, the particle filter is capable of tracking the targets, but the set RMSE performance still suffers compared to the EKF and UKF. This can be attributed to a problem with the likelihood density. Repeating the analysis from Section 2.4, imagine three-dimensional axes as in Figure 13. On two axes are the "x" Cartesian positions, x_1 and

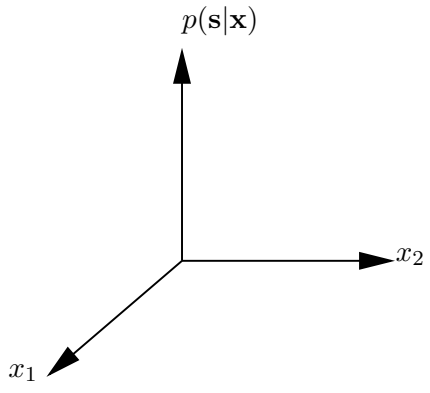


Figure 13: Coordinate axes for visualization of the two-dimensional SME likelihood function.

Table 11: Associated Root Mean Squared Error

	Sum of Products	Sum of Powers
Extended Kalman	3.0	3.0
Unscented Kalman	3.0	3.0
PF - Analytical	4.9	5.0
PF - Unscented	5.0	5.0
Global NN	3.0	
Associated KF	3.0	

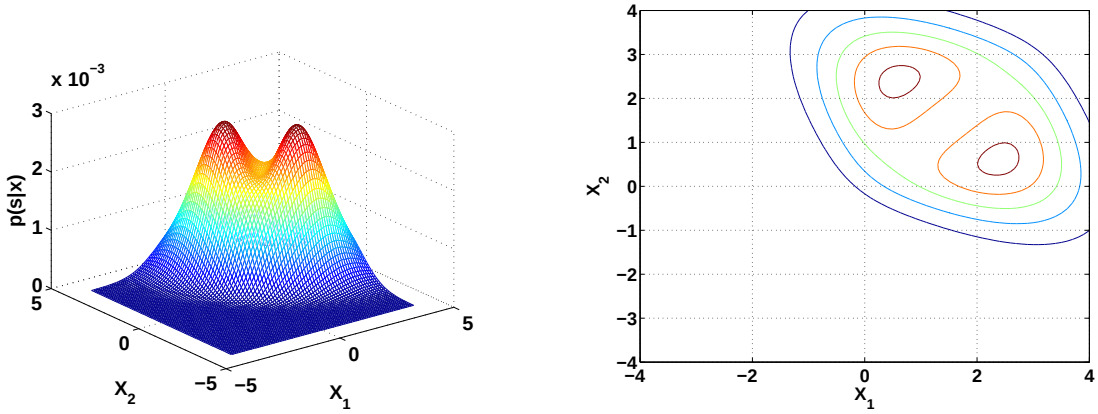
x_2 for two of the targets. On the third axis is the likelihood of the SME pseudomeasurement given the state, $p(\mathbf{s}|\mathbf{x})$, where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix}. \tag{110}$$

To simplify analysis, assume $y_1 = y_2 = 0$. To visualize the likelihood, imagine a grid of uniformly spaced "particles" around an observation

$$\begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{111}$$

Figure 14 shows the likelihood as a function of x_1 and x_2. We expect to see peaks in the likelihood at $(3, 0)$ and $(0, 3)$, since the SME does not distinguish between these two values. There are two peaks in the likelihood, but they appear at $(2.4, 0.6)$ and $(0.6, 2.4)$ rather than at $(3, 0)$ and $(0, 3)$, which is more vexing than in the one-dimensional case where, in a similar situation, the two peaks occur at $(2.7, 0.3)$ and $(0.3, 2.7)$. With this likelihood, the particle filter will give more weight to a particle at $(2.4, 0.6)$ than one at the actual location of the observation. This is a result of using Gaussian approximations to the likelihood, and the inaccuracies in these approximations generate problems that adversely affect the performance of the particle filter. Given the poor RMSE performance and high



(a) EKF approximation around $(3, 0, 0, 0)$.

(b) EKF approximation around $(3, 0, 0, 0)$.

Figure 14: "Likelihood" for two targets moving in two dimensions; Sum of Products SME.

computational complexity of the particle filter as implemented here, it is clearly inferior to other approaches.

3.7 Conclusions

This chapter presented several new, but discouraging, results. First, we presented a new set of two-dimensional SME equations based on the sum-of-powers one-dimensional SME. In a relatively easy scenario, the new sum-of-powers SME performed as well as the sum-of-products implementation, but it performed poorly in a more challenging scenario. Second, we combined the UKF and particle filter with SME in two-dimensional tracking. Contrary to expectations, the UKF and particle filter performed worse than the EKF. The UKF experienced problems maintaining positive definite covariance estimates, and the approximations made to implement the particle filter degraded its performance too much for the particle filter to be useful. These results, contrary to initial expectations, suggest that the EKF may be the best nonlinear filter choice for SME-based tracking in two dimensions.

CHAPTER IV

A NOVEL EXTENSION OF SME TRACKING TO THREE-DIMENSIONAL MOTION

As described in the previous two chapters, SME tracking for targets moving with one-dimensional and two-dimensional motion is well established. However, no satisfactory method for SME tracking in three dimensions has previously been presented. This chapter rectifies that situation.

4.1 Background

4.1.1 Previous State of the Art

While two approaches for three-dimensional tracking have been proposed [24,31], neither of these methods is particularly attractive. One solution is to separate the tracking problem into two parts – a two-dimensional problem that can be solved with the complex number approach described in Chapter 3 and a one-dimensional problem. While better than separating the problem into three independent one-dimensional problems, this approach requires the targets to be well-separated in at least one dimension. As mentioned in the previous chapter, [31] presents another solution for performing N-dimensional SME-based tracking without the coordinate-switching/ghost-target problem. Their approach encodes different coordinates of a measurement as coefficients of a polynomial. The downside to their technique is that it requires a larger number of SME equations ($N^2 + 2N$ for N targets in three dimensions) than other approaches ($3N$) and consequently has higher complexity and computational resource needs. Our approach, based upon an extension of the complex numbers known as quaternions, possesses a combination of positive features not exhibited by these existing three-dimensional SME approaches.

4.1.2 Quaternions

Intrigued by the elegance and simplicity of the complex-number approach to SME-based two-dimensional tracking, we searched for something similar that would allow three-dimensional tracking. This search lead us to quaternions. Developed by Sir William Rowan Hamilton in the middle 19^{th} century, quaternions are a non-commutative extension of the complex numbers. While eventually surpassed in most applications by vector notation, quaternions are still commonly used as an efficient method for representing and interpolating rotations in applications such as computer graphics [45] and line-of-sight control [1]. A quaternion q is written as

$$q = a + bi + cj + dk, \tag{112}$$

where i, j, and k are "imaginary" variables that satisfy the equation

$$i^2 = j^2 = k^2 = -1. \tag{113}$$

Quaternion addition is defined as expected from complex numbers, but multiplication of the quaternion bases is defined as

$$ij = k, \quad ji = -k, \tag{114}$$

$$jk = i, \quad kj = -i, \tag{115}$$

$$ik = -j, \quad ki = j. \tag{116}$$

The main problem with quaternions is their lack of commutativity in multiplication, i.e. $ab \neq ba$. At first, this might seem to eliminate them from consideration for forming SMEs, since the SME approach is based on the idea that order should not matter. However, the following section will show how it is possible to use quaternions for three-dimensional SME-based tracking if the representation and SME equations are chosen carefully.

4.2 Quaternion-based SME

4.2.1 Choosing the Quaternion Representation

In choosing a quaternion-based SME implementation, there are two areas where we have design flexibility: the quaternion encoding and the SME themselves. We have three pieces

of information for each target, namely the x, y, and z Cartesian coordinates, but four places where we can encode this information in a quaternion. Initially, the most appealing choice might be to use the three imaginary parts

$$q = 0 + xi + yj + zk, \tag{117}$$

as is done when using quaternions for rotations [45]. However, choosing the quaternion as

$$q = x + yi + zj + 0k \tag{118}$$

works better for SME implementation, as we will see in the next section.

4.2.2 Choosing the Set of SME Equations

The choices for the SME that we consider are the sum-of-products and the sum-of-powers forms. We first examine the sum-of-products implementation.

4.2.2.1 Sum of Products

We will begin with the simplest case – two targets moving in three dimensions. In this case, the set of SME equations is given by

$$\tilde{m}_{qprod} = \begin{pmatrix} \mathcal{R}\{q_1 + q_2\} \\ \mathcal{I}_i\{q_1 + q_2\} \\ \mathcal{I}_j\{q_1 + q_2\} \\ \mathcal{R}\{q_1 q_2\} \\ \mathcal{I}_i\{q_1 q_2\} \\ \mathcal{I}_j\{q_1 q_2\} \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ y_1 + y_2 \\ z_1 + z_2 \\ x_1 x_2 - y_1 y_2 - z_1 z_2 \\ x_1 y_2 + x_2 y_1 \\ x_1 z_2 + x_2 z_1 \end{pmatrix}, \tag{119}$$

where q is a quaternion representing a three-dimensional position encoded as in (118). $\mathcal{I}_i$ and $\mathcal{I}_j$ represent the i-part and j-part of the quaternion. Notice that all the equations on the right hand side of (119) have the symmetric property. We do not use the k-part of the quaternion. For $q_1 + q_2$, this works out naturally as the k-part is zero anyway. This extends to $q_1 + q_2 + q_3$, and so on, for a higher number of targets. However, for $q_1 q_2$, the k-part would be $y_1 z_2 - z_1 y_2$, which is not symmetric. Because we already have six equations, we

probably can ignore the k-part, but we first need to check one restriction. A proper set of SME equations must have the property that the original measurements are recoverable from SME pseudomeasurements, up to a permutation [18, 19]. Ignoring the k-part may raise concern over whether this requirement is still met. We return to the Implicit Function Theorem (IFT), introduced in Section 3.2.1, to show that this requirement is met. The Jacobian matrix, $\mathcal{J}$, for the three-dimensional two-target sum-of-products SME is given by

$$\mathcal{J}_{prod} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ x_2 & -y_2 & -z_2 & x_1 & -y_1 & -z_1 \\ y_2 & x_2 & 0 & y_1 & x_1 & 0 \\ z_2 & 0 & x_2 & z_1 & 0 & x_1 \end{pmatrix}, \tag{120}$$

and its determinant is

$$det(\mathcal{J}) = (x_1 - x_2)\left[(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2\right]. \tag{121}$$

From this equation, we can see that the necessary inverse functions exist everywhere except when the targets have the same position in the x-coordinate. In both cases, we expect the number of "independent" equations to be reduced and it can be shown, using the IFT, that the required number of "independent" equations still exist.

Extending the sum-of-products to three targets, however, runs into an immediate problem, since it requires a $q_1q_2q_3$ term. Unfortunately, only the real part of this term is symmetric, leaving us only seven symmetric equations instead of the nine we need. Hence, this sum-of-products formulation is apparently only useful for two targets.[1] Fortunately, there is another SME at our disposal that may be useful for a higher number of targets.

[1] While it may be possible to tweak the sum-of-products implementation to get around this issue, we have not yet discovered any such method.

4.2.2.2 Sum of Powers

The equation for the sum-of-powers quaternion implementation for two targets moving in three dimensions is

$$\tilde{m}_{qprod} = \begin{pmatrix} \mathcal{R}\{q_1 + q_2\} \\ \mathcal{I}_i\{q_1 + q_2\} \\ \mathcal{I}_j\{q_1 + q_2\} \\ \mathcal{R}\{q_1^2 + q_2^2\} \\ \mathcal{I}_i\{q_1^2 + q_2^2\} \\ \mathcal{I}_j\{q_1^2 + q_2^2\} \end{pmatrix} \Rightarrow \begin{pmatrix} x_1 + x_2 \\ y_1 + y_2 \\ z_1 + z_2 \\ x_1^2 - y_1^2 - z_1^2 + x_2^2 - y_2^2 - z_2^2 + 2\sigma^2 \\ 2x_1y_1 + 2x_2y_2 \\ 2x_1z_1 + 2x_2z_2 \end{pmatrix}. \tag{122}$$

Looking at the expanded version on the right-hand side of (122), we see that this formulation is symmetric. Examining the left-hand side of (122), the sum-of-powers SME avoids the non-commutativity of quaternions altogether. Non-commutativity only comes into play when two *different* quaternions are multiplied. However, in the sum-of-powers SME, quaternions are only multiplied with themselves, so non-commutativity is not an issue. The second convenient property of the sum-of-powers implementation is that the k-part is naturally zero; it is easy to show that this is true for an arbitrary number of targets. One minor complication with the sum-of-powers implementation is the $2\sigma^2$ factor needed in the fourth equation of the set. This factor is there to ensure that the process is zero-mean. In writing this correction factor as such, we have assumed that all the x_i's, y_i's, and z_i's are each disturbed by additive, independent white Gaussian noises distributed according to $\mathcal{N}(0, \sigma^2)$. This identical-distribution assumption can be relaxed, but it would lead to a more cluttered set of equations. It is clear that the sum-of-powers implementation can be readily expanded to any number of targets. Finally, we present the determinant of the Jacobian matrix to demonstrate that the inverse set of SME equations do exist by the IFT:

$$det(\mathcal{J}) = -8(x_1 - x_2)\left[(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2\right]. \tag{123}$$

Surprisingly, this is identical to the sum-of-products case (121), up to a scalar constant, and consequently, it meets the requirements exactly as the sum-of-product form did.

4.3 Simulations

Implementation of SME-based tracking requires a nonlinear filter. For this example, we have chosen the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) [16, 17]. The EKF demonstrated good performance for one and two-dimensional SME tracking (Chapters 2 and 3), and the UKF offers easy implementation. To implement the SME-based tracking filter using an EKF, the Jacobian matrix of the SME and a Gaussian approximation for the observation noise need to be found. We derive these formulas for the sum-of-products and the sum-of-powers in Appendix C. Examining the equation for the Gaussian approximation makes the appeal of the UKF clear, since it does not require these cumbersome calculations.

Figure 15 shows an example scenario of two targets moving in three-dimensional space. The motion is assumed to follow a constant velocity model with independent zero-mean process noises in each dimension. Figure 16 shows another scenario designed to stress the

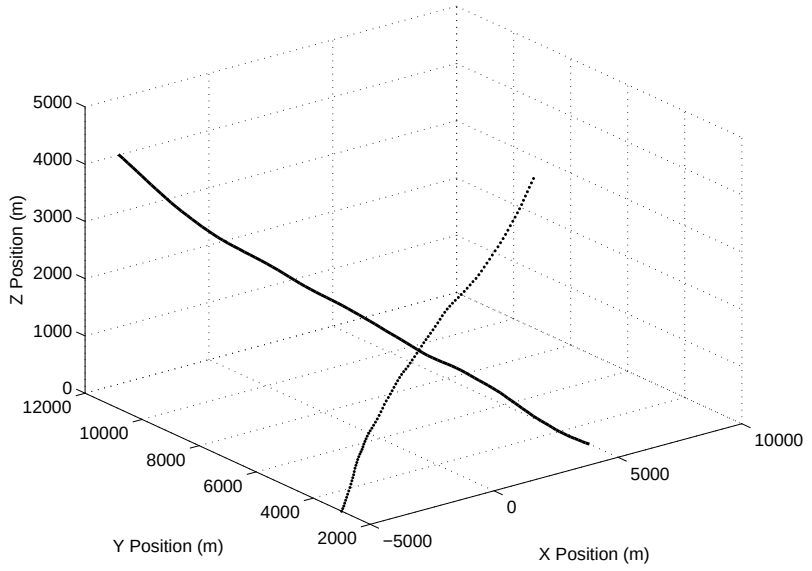


Figure 15: Scenario 1 – Two targets moving in three dimensions.

filter. The two targets start close together (lower left hand corner of figure) and move in nearly parallel paths.

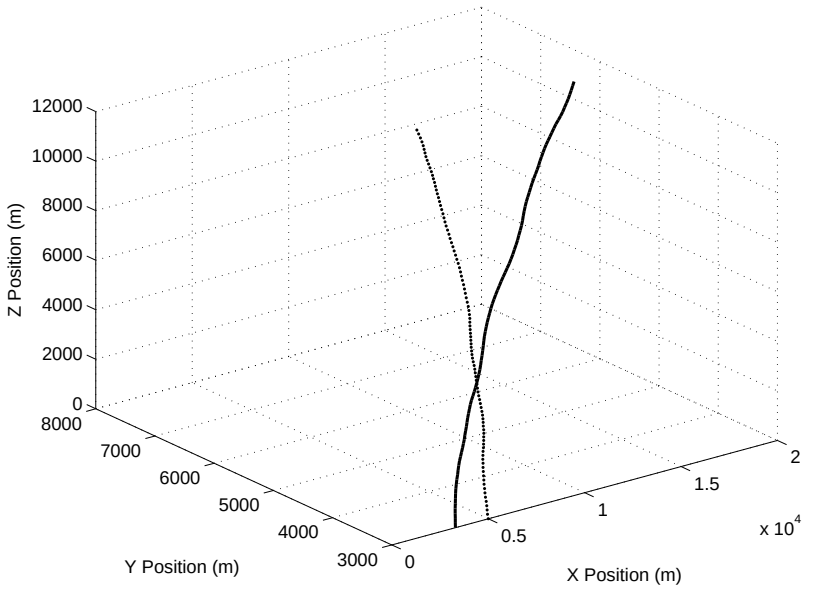


Figure 16: Scenario 2 – Two targets moving in three dimensions.

4.3.1 Data

Three types of results for these two scenarios are presented in the following tables. For this set of results, the observation noise variance was set to 1×10^4 m^2. This high value was chosen to induce the tracking filters to break at least a small percentage of the time. The process noise variance was set to 10 m^2/s^4, and 1000 Monte Carlo trials were performed. Results for a global nearest neighbor (GNN) data assignment algorithm feeding two Kalman filters and an associated[2] Kalman filter are also given in the tables for comparison.

Tables 12 and 13 show the percentage of correctly[3] maintained tracks for Scenarios 1 and 2, respectively. Tables 14 and 15 give the average RMSE.[4] Only simulation runs in which the filter maintained correct target associations are included in this calculation. Tables 16 and 17 give the "set estimation" RMSE. The set estimation RMSE is calculated by finding the track-estimate/track-truth association with the smallest MSE at each time step and then averaging over all time steps. Consequently, target associations are not considered.

[2]The "associated" Kalman filter is actually two Kalman filters running independently on the two tracks. Data assignment is not an issue since we know the correct association in the simulation.

[3]An incorrectly maintained track means that the track for target 1 began following target 2 at some point and vice versa.

[4]RMSE is averaged over time and Monte-Carlo trials.

Table 12: Percentage of Trials with Correctly Maintained Associations - Scenario 1

	3D Sum of Products	3D Sum of Powers
Extended Kalman	100	100
Unscented Kalman	100	100
Global NN	100	
Associated KF	100	

Table 13: Percentage of Trials with Correctly Maintained Associations - Scenario 2

	3D Sum of Products	3D Sum of Powers
Extended Kalman	94.8	89.1
Unscented Kalman	90.3	69.5
Global NN	97.2	
Associated KF	100	

Table 14: Associated Root Mean Squared Error - Scenario 1

	3D Sum of Products	3D Sum of Powers
Extended Kalman	84.0	84.2
Unscented Kalman	99.4	94.7
Global NN	83.8	
Associated KF	83.7	

Table 15: Associated Root Mean Squared Error - Scenario 2

	3D Sum of Products	3D Sum of Powers
Extended Kalman	92.1	97.5
Unscented Kalman	437.1	622.8
Global NN	84.0	
Associated KF	83.5	

Table 16: Set Estimation Root Mean Squared Error - Scenario 1

	3D Sum of Products	3D Sum of Powers
Extended Kalman	84.0	84.2
Unscented Kalman	99.4	94.7
Global NN	83.8	
Associated KF	83.7	

Table 17: Set Estimation Root Mean Squared Error - Scenario 2

	3D Sum of Products	3D Sum of Powers
Extended Kalman	92.8	98.7
Unscented Kalman	490.3	604.4
Global NN	84.1	
Associated KF	83.5	

Track switching was never a problem in Scenario 1, which is not surprising since the targets almost never cross through the same point at the same time. For Scenario 1, the SME methods implemented with the EKF show slightly worse RMSE performance than the GNN filter, and the UKF-based implementations have noticeably worse RMSE performance.

Scenario 2 was designed to stress the filters and the results bear this out. The filtering algorithms now occasionally switch tracks. The GNN approach has the lowest occurrence, followed closely by the EKF. The UKF is essentially useless in this case. This is probably a consequence of the instability of the UKF when estimates of the targets' positions cross (See Chapter 2.1.3, [26, 27]). An example of the UKF breaking down can be seen in Figure 17. The same performance ordering occurs for both RMSE metrics. The final observation is that the sum-of-products SME, coupled with the EKF, seems to provide slightly better performance than the sum-of-powers form coupled with the EKF; this is consistent with our observations in the one-dimensional and two-dimensional cases.

4.4 Conclusion

This chapter presented a new method for efficiently extending the SME multiple-target tracking method to targets moving in three dimensions. Through creative use of quaternions, we may track in three dimensions with the minimal possible number of SME equations while avoiding the coordinate-switching problem that potentially appears in other methods that also meet this minimum equation property. Repeated Monte Carlo trials of the method demonstrate its viability, although more thorough testing would be required to determine its applicability as a tracking algorithm. However, our primary interest in SME methods do not lie in their applicability to tracking, but instead as tactics for computing Cramér-Rao

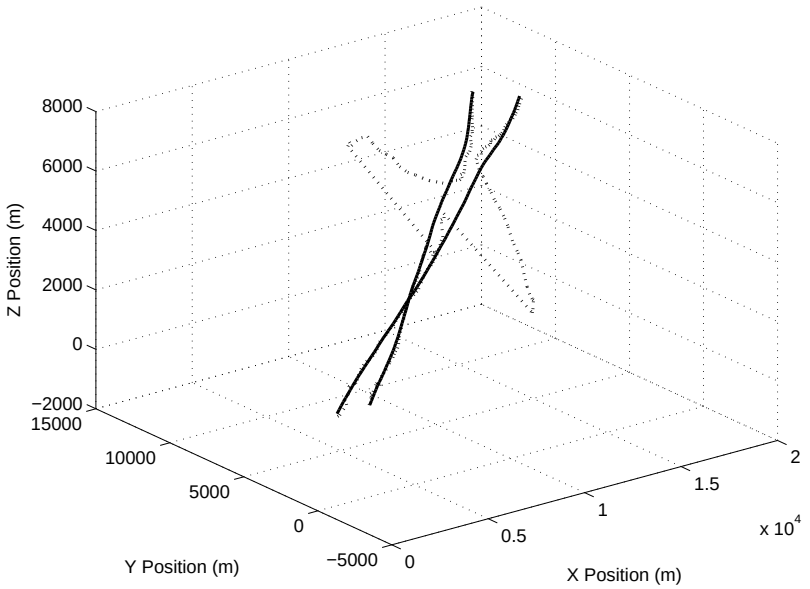


Figure 17: Illustration of the UKF malfunctioning – Scenario 2.

bounds for multiple target tracking, as suggested in [10] or [38]. This will be considered in the next chapter.

CHAPTER V

APPROXIMATE CRAMÉR-RAO BOUNDS FOR MULTIPLE TARGET TRACKING

Thus far, we have discussed methods for using symmetric measurement equations to track targets in one, two, and three dimensions. Our results suggest that the SME approach, as currently implemented, may not be well-suited as a general multiple target tracking method. It is only applicable for a small number of targets, and other data association methods, such as global nearest neighbor assignment, perform as well, if not better, in these situations. Even the application of nonlinear filtering techniques newer than the EKF has not generally improved SME performance. Although future researchers may discover modifications of our algorithms (perhaps alternate choices of SME, better likelihood approximations, and other nonlinear filters) that improve tracking performance, we currently have another application for SME in mind: the computation of approximate Cramér-Rao bounds for multiple target tracking.

5.1 Background

Optimal solutions to multiple target tracking (MTT) problems are intractable in most situations of interest. Although many suboptimal solutions to MTT problems have been proposed, system designers have few tools at their disposal, other than trial-and-error, for predicting the performance of MTT systems. One common solution to similar problems has been to develop bounds on the performance of a filtering system, such as the Cramér-Rao lower bound (CRLB). The CRLB provides a lower bound on the variance of an unbiased

estimator. Specifically, the CRLB for vector parameters can be written as

$$\begin{aligned} \mathrm{var}(T(\mathbf{z})) &\geq J^{-1}(\mathbf{x}), \\ J(\mathbf{x}) &= E\left[(\nabla_{\mathbf{x}} \ln p(\mathbf{z};\mathbf{x}))(\nabla_{\mathbf{x}} \ln p(\mathbf{z};\mathbf{x}))^T\right], \\ J(\mathbf{x}) &= -E\left[\nabla_{\mathbf{x}}\{(\nabla_{\mathbf{x}} \ln p(\mathbf{z};\mathbf{x}))^T\}\right]. \end{aligned} \tag{124}$$

where

$\mathbf{x}$	$r \times 1$ unknown deterministic vector parameter,
$\mathbf{z}$	$r \times k$ measurement matrix (k measurements),
T	estimator,
J	Fisher information matrix.

Recognizing that

$$\mathrm{var}(T(\mathbf{z})) = MSE(T(\mathbf{z})) = E\left[(T(\mathbf{z}) - \mathbf{x})(T(\mathbf{z}) - \mathbf{x})^T\right], \tag{125}$$

the CRLB can be interpreted as a lower bound on the mean squared error (MSE) of an estimator.

It turns out that even bounds are quite difficult to compute for MTT problems. In [10], Daum suggested that the symmetric measurement equation (SME) approach to MTT, studied in the previous chapters, could be used to efficiently compute a bound. This chapter discusses our findings from implementing that idea. While producing a true bound is not practical, the resulting method serves as a performance prediction tool. Designers may vary process and observation noises as well as the number and paths of the targets to determine how an MTT system might respond. The key to the approach described in this chapter is to recognize that the SME tactic converts the MTT problem into a nonlinear estimation problem. Existing techniques for computing nonlinear estimation bounds can then be applied to the MTT problem.

5.2 CRLB for Target Tracking

While a number of nonlinear estimation bounding techniques exist, the one that most easily solves our problem is the method suggested by Taylor [36]:

> "the inverse information matrix (P*) corresponding to a dynamic system modelled by a nonlinear time-varying state vector differential equation with deterministic inputs and nonlinear time-varying observations of the state variables corrupted by additive Gaussian white noise sequences propagates according to the same equations as the filter covariance matrix for an extended Kalman filter (EKF) linearized about the true (unknown) trajectory."

In essence, Taylor says that by replacing an EKF's estimate with the truth at each time step, the tweaked EKF recursively computes the CRLB under the right circumstances. His original formulation allows for a nonlinear state update function and for a series of *known* deterministic inputs, $\mathbf{u}_k$. Let our system be defined by the following equations, with the subscript indicating the time index:

$$\mathbf{x}_{k+1} = \Phi\mathbf{x}_k + \Gamma\mathbf{w}_k + \mathbf{u}_k, \tag{126}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k, \tag{127}$$

where

$\mathbf{x}_k$	state at time k,
$\mathbf{u}_k$	known deterministic input at time k,
$\mathbf{z}_k$	measurement at time k,
$\mathbf{w}_k$	white Gaussian process noise $\sim \mathcal{N}(\mathbf{0}, Q)$,
$\mathbf{v}_k$	white Gaussian observation noise[1] $\sim \mathcal{N}(\mathbf{0}, R)$,
Φ	constant velocity state update matrix,
h_k	SME observation equation.

With this system, the state-update EKF equations are given by

$$\hat{\mathbf{x}}_k^- = \Phi\hat{\mathbf{x}}_{k-1}, \tag{128}$$

$$P_k^- = \Phi P_{k-1}\Phi^T + \Gamma Q\Gamma^T, \tag{129}$$

where a superscript $^-$ indicates an *a priori* value, P_k^- is the *a priori* estimation error covariance matrix, and P_k is the *a posteriori* estimation error covariance matrix. The EKF

[1] In SME tracking, $\mathbf{v}_k$ is additive and white, but not Gaussian. However, we generally approximate it as Gaussian.

measurement update equations are given by

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R]^{-1}, \tag{130}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k[\mathbf{z}_k - H_k \hat{\mathbf{x}}_k^-], \tag{131}$$

$$P_k = P_k^- - K_k H_k P_k^-, \tag{132}$$

where H_k is the Jacobian matrix for the set of SME functions at time k.

Taylor shows, in [36], that the inverse Fisher information matrix for the system described in (126)-(127) propagates recursively as

$$J(\mathbf{x}_k) = (\Phi^{-1})^T J(\mathbf{x}_{k-1}) \Phi^{-1} + H_k^T R_k^{-1} H_k. \tag{133}$$

Furthermore, he recognized that the general EKF computes the inverse of the filter covariance matrix according to the recursive equation

$$P_k^{-1} = [\Phi P_{k-1} \Phi^T + Q]^{-1} + H_k^T R_k^{-1} H_k, \tag{134}$$

if the true state trajectory, $\mathbf{x}$, replaces the estimated one, $\hat{\mathbf{x}}$.[2] By making the additional assumption that $Q = 0$ (i.e., process noise does not exist), (134) becomes

$$P_k^{-1} = (\Phi^{-1})^T P_{k-1}^{-1} \Phi^{-1} + H_k^T R_k^{-1} H_k. \tag{135}$$

Comparing (133) and (135),

$$J(\mathbf{x}_k) = P_k^{-1}, \tag{136}$$

$$J(\mathbf{x}_k)^{-1} = P_k. \tag{137}$$

Recalling that the Cramér-Rao lower bound for an unbiased estimator states that

$$P_{estimator,k} \geq J(\mathbf{x}_k)^{-1}, \tag{138}$$

we now have a simple recursive method for computing the CRLB.[3] Notice that Taylor's bound is a function of a specific path $\mathbf{x}$, but it assumes that the deterministic process input $\mathbf{u}_k$ is known. Of course, in a real system, this process is typically modeled with process noise.

[2] The derivation of this fact is a tedious and sometimes tricky exercise in linear algebra. The details are worked out in Appendix D.

[3] $A \geq B$ means that $A - B$ is nonnegative definite.

5.2.1 The UKF for Bound Computation

One goal of this work was to compute bounds using the unscented Kalman filter (UKF) instead of the EKF. The SME approach takes the observations, which are assumed to be the truth plus additive Gaussian noise, and applies a set of nonlinear equations to them. Consequently, the observation noise becomes non-Gaussian. Earlier publications on EKF/SME-based tracking present methods for finding an approximation for this non-Gaussian observation noise [19, 20, 27, 35]. These approximations are, however, tedious to calculate and constitute an impediment to EKF-based SME implementations. The UKF also uses a Gaussian approximation for the observation noise, but it generates that approximation automatically, which significantly eases the development of SME-based tracking algorithms or performance prediction techniques. The appeal of the UKF is clear, but how valid is the UKF for computing bounds?

In Chapter 2, we explored Taylor series analysis of the UKF. From that analysis, we know that the EKF truncates the Taylor series after the first term. The UKF, on the other hand, matches the Taylor series through the first four terms of a symmetric distribution, and,in general, is more accurate than the truncation employed in the EKF. With this in mind, it is reasonable to hope that the UKF-based bounding technique will perform at least as well as, if not better than, the EKF-based one. However, we will see that the singularity issue experienced when using the UKF for SME tracking haunts bound computation with the UKF as well.

5.2.2 Limitations of Combining the SME with Taylor's CRLB

Both Taylor's EKF and the UKF bounding technique require two assumptions. The first is that observation noise is additive white Gaussian noise (AWGN). While both the EKF and UKF approaches to SME tracking model the observation noise as AWGN, the noise is not AWGN because the SME formulation involves nonlinear operations. To see this, suppose our system is tracking N targets, and let the actual received measurement, m_i, be modeled as the truth, x_i, plus an additive Gaussian noise, n_i, so that

$$m_i = x_i + n_i \qquad \text{for } i = 1, \ldots, N. \tag{139}$$

Applying an SME observation function $g(\cdot)$ to the actual measurements yields

$$z_i = g_i(m_1, m_2, \cdots, m_N) \quad \text{for } i = 1, \ldots, N. \tag{140}$$

The restrictions on the choice of SME functions given in [18] require that (140) can be written as

$$z_i = g_i(x_1, x_2, \cdots, x_N) + v_i \quad \text{for } i = 1, \ldots, N, \tag{141}$$

where v_i is a zero mean white noise term whose distribution depends on $x_1, x_2, \cdots, x_N$. Because the SME functions are nonlinear, v_i is not Gaussian, and one of the assumptions is violated.

The second assumption is that there is no process noise, which guarantees that the state variables are deterministic rather than another random process. However, for target tracking, this assumption is too limiting because trackers almost never follow targets moving with deterministic motion.

5.3 Posterior Cramér-Rao Bound

Van Trees, aware of the deterministic parameter limitation of the CRLB, introduced the posterior Cramér-Rao bound (PCRB) [40]. Let $T(\mathbf{z}_k)$ be an estimator of the $r \times 1$-dimensional state, $\mathbf{x}_k$. The PCRB is then given by

$$\text{var}(T(\mathbf{Z}_k)) \geq J^{-1}, \tag{142}$$

$$J = E\left[(\nabla_{\mathbf{x}} \ln p(\mathbf{Z}_k, \mathbf{X}_k))(\nabla_{\mathbf{x}} \ln p(\mathbf{Z}_k, \mathbf{X}_k))^T\right], \tag{143}$$

$$J = -E\left[\nabla_{\mathbf{x}}\{(\nabla_{\mathbf{x}} \ln p(\mathbf{Z}_k, \mathbf{X}_k))^T\}\right], \tag{144}$$

where

$$\mathbf{X}_k = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k\}, \tag{145}$$

$$\mathbf{Z}_k = \{\mathbf{z}_0, \mathbf{z}_1, \cdots, \mathbf{z}_k\}, \tag{146}$$

and J is now a Bayesian posterior Fisher information matrix. Note the expectation in (143) and (144) is taken over both $\mathbf{X}_k$ and $\mathbf{Z}_k$. This differs from the CRLB (and Taylor's method of computing it), where $\mathbf{X}_k$ is a deterministic parameter and the expectation is only over $\mathbf{Z}_k$.

Consequently, the Taylor bound is a function of a specific *deterministic* path $\mathbf{X}_k$. However, the PCRB averages over possible paths weighted by their prior probability.[4] Computation of the PCRB by batch processing of the data, as described in (142), quickly becomes intractable for a target tracking scenario where the time k can grow large. Fortunately, Tichavský *et al.* introduced a recursive method of finding the PCRB for the general nonlinear filtering problem [38]. Consider the general nonlinear filtering formulation

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{w}_k, \mathbf{u}_k), \tag{147}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k, \mathbf{n}_k), \tag{148}$$

where

$\mathbf{x}_k$	state at time k,
$\mathbf{u}_k$	known deterministic input at time k,
$\mathbf{z}_k$	measurement at time k,
$\mathbf{w}_k$	white process noise,
$\mathbf{n}_k$	white observation noise independent of $\mathbf{w}_k$,
f	nonlinear state update equation,[5]
h_k	nonlinear observation equation; may vary with time.

Using (147) and (148) along with an initial state distribution $p(\mathbf{x}_0)$, the joint probability density of $\mathbf{X}_k$ and $\mathbf{Z}_k$ can be rewritten as

$$p(\mathbf{Z}_k, \mathbf{X}_k) = p(\mathbf{x}_0) \prod_{i=1}^{k} p(\mathbf{z}_i|\mathbf{x}_i) \prod_{j=1}^{k} p(\mathbf{x}_j|\mathbf{x}_{j-1}). \tag{149}$$

Examining the structure of (149), Tichavský *et al.* realized that a recursive computation of the PCRB might be possible, and they presented such an algorithm in [38]. We summarize the derivation here; see [38] for details. Suppose $\mathbf{x}_k$ is an r-dimensional vector, and let J represent the $(rk \times rk)$ Fisher information matrix (FIM) of $\mathbf{X}_k$. Denote the $(r \times r)$ FIM for $\mathbf{x}_k$ as J_k. Defined in this manner, the sequence $\{J_1, J_2, \cdots, J_k\}$ is the sequence of posterior information matrices for estimating the sequence of state vectors $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}$. J_k can be computed recursively according to

$$J_{k+1} = D_k^{22} - D_k^{21}(J_k + D_k^{11})^{-1} D_k^{12}, \tag{150}$$

[4]The prior probability is determined by the process noise statistics.

[5]In [38], f is a possibly time varying function, but the deterministic input $\mathbf{u}_k$ does not exist. For comparison to the Taylor derivation and to later clarify what seems to be common practice in the literature, we have removed f's time dependence and added $\mathbf{u}_k$ as an input.

where

$$D_k^{11} = E\left[-\nabla_{\mathbf{x}_k}\nabla_{\mathbf{x}_k}^T p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right], \tag{151}$$

$$D_k^{12} = E\left[-\nabla_{\mathbf{x}_k}\nabla_{\mathbf{x}_{k+1}}^T p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right], \tag{152}$$

$$D_k^{21} = [D_k^{12}]^T, \tag{153}$$

$$D_k^{22} = E\left[-\nabla_{\mathbf{x}_{k+1}}\nabla_{\mathbf{x}_{k+1}}^T p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right], \tag{154}$$

$$+E\left[-\nabla_{\mathbf{x}_{k+1}}\nabla_{\mathbf{x}_{k+1}}^T p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})\right].^6 \tag{155}$$

Relating the PCRB back to the situation discussed in this work, the SME is the nonlinear measurement function $h(\mathbf{x}_k)$. These observations are corrupted by additive white noise $\mathbf{v}_k$, which is an approximation to the noise in $h(\mathbf{x}_k, \mathbf{n}_k)$ as discussed in Section 5.2.2. The state transition function, $f(\mathbf{x}_k)$, is a linear constant velocity model perturbed by additive Gaussian process noise, so (147) can be rewritten as

$$\mathbf{x}_{k+1} = \Phi\mathbf{x}_k + \Gamma\mathbf{w}_k + \mathbf{u}_k. \tag{156}$$

We would be ready to use this algorithm to compute the PCRB, except for one problem. Evaluation of D_k^{22} requires knowledge of $p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})$. With the SME approach, this is the same density Section 5.2.2 describes as being extremely difficult to find. Hence, we use a Gaussian approximation to the SME data density.

5.3.1 Linearizing the PCRB

Recalling one of the engineer's favorite tricks, we can linearize $h(\mathbf{x}_k)$ as suggested in [37]

$$H_k = \left.\frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k=\hat{\mathbf{x}}_k}, \tag{157}$$

where $\hat{\mathbf{x}}_k$ is the known *average* target path[7] the bound is being calculated for. In the literature, researchers often plug in a specific, rather than an average, path; some authors seem to imply they are computing a conditional PCRB conditioned on that specific path. We believe this interpretation is erroneous, and the described procedure actually computes

[6] The expectations in (151)-(155) are conditional expectations.

[7] The *average* path is one that follows the state model exactly, including the deterministic inputs $\mathbf{u}_k$, with no random perturbations $\mathbf{w}_k$.

a bound that assumes the process noise is a *deterministic* input $\mathbf{u}_k$ that is known by the tracking filter. Continuing the derivation of the linearized PCRB, assume the observation and process noises are additive white Gaussian with distributions $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q_k)$ and $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, R_k)$, respectively. Then, the authors of [37] show that (151)-(155) simplify to

$$D_k^{11} = \Phi_k^T Q_k^{-1} \Phi_k, \tag{158}$$
$$D_k^{12} = -\Phi_k^T Q_k^{-1}, \tag{159}$$
$$D_k^{21} = -Q_k^{-1} \Phi_k, \tag{160}$$
$$D_k^{22} = Q_k^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}. \tag{161}$$

Substituting (158)-(161) into (150),

$$J_{k+1} = H_{k+1}^T R_{k+1}^{-1} H_{k+1} + (Q_k + \Phi_k J_k \Phi_k^T)^{-1}. \tag{162}$$

Comparing (162) to (135), it is apparent that these two equations are *identical*. This is not necessarily surprising since the EKF is based on linearizations, but it does lead to an interesting conclusion: Taylor's method for computing the CRLB can be easily made to compute an approximate PCRB by simply including the process noise covariance.

5.4 Comparing Monte Carlo Trials to Bounds

Recall that the bound we are trying to find is a bound on the estimator (tracker) mean squared error (MSE) at each time step k:

$$MSE(T(\mathbf{z_k})) = E\left[(T(\mathbf{z_k}) - \mathbf{x_k})(T(\mathbf{z_k}) - \mathbf{x_k})^T\right]. \tag{163}$$

It seems straightforward to run N Monte Carlo trials where a practical tracking algorithm estimates the targets' positions and compute a sample MSE

$$MSE(T(\mathbf{z_k})) \approx \frac{1}{N} \sum_{n=1}^{N} \left[(T(\mathbf{z_k}) - \mathbf{x_k})(T(\mathbf{z_k}) - \mathbf{x_k})^T\right]. \tag{164}$$

Assuming that the tracker is unbiased, our sample MSE should be bounded by the CRLB if the MSEs above are computed with $\mathbf{x_k}$ as a deterministic parameter and bounded by the PCRB if $\mathbf{x_k}$ is a random parameter that the PCRB averages over. Practical implementation of the sample MSE runs into two immediate problems, however. First, how should the MSE

be computed for multiple target tracking? Second, what target paths should the bound be computed around and tracking algorithms run on? The following sections will try to answer these questions.

5.4.1 Computing the MSE

Suppose that there are two (or more) targets in the scenario and their paths cross. How should error be calculated if the tracking filter switches tracks (i.e., after the targets cross, the filter thinks target 1 is target 2 and vice versa)? The first option is to consider this an error and compute the MSE accordingly. However, if this option is chosen, even a single target switch will ruin the result as the MSE becomes enormous, making our lower bounds seem trivial. A second solution is to reject any simulation run where the filter confuses tracks. This solution is unsatisfactory because difficult scenarios are thrown out. The final option is to consider a set error. In this case, track associations are not considered important. As long as the filter maintains a track for every target, it is considered to be working. The MSE at each time instant is then computed as the error between a target and the nearest track. This is the MSE calculation we have chosen to use. Of course, none of the bounds we have described know about this "set error," so we do not necessarily expect them to be true lower bounds on the set error; however, we may conjecture that they will follow the same trends.

5.4.2 Comparison for the CRLB

Now for the second question: which target paths should be used for the bound computation and Monte Carlo trials? As discussed earlier, the CRLB requires that the targets' paths be deterministic. This makes the path choice simple for the CRLB case. Throughout this work, we assume a constant velocity model, so we generate a set of tracks where all the targets move exactly according to this model (no process noise). We compute an approximate CRLB around this noiseless track using Taylor's method. The SME handles the data association and feeds the resulting pseudomeasurements into an EKF or UKF. For the Monte Carlo trials, a different realization of the observation noise is added for each trial. A tracking algorithm is then run on the noisy tracks and then the sample MSE is computed.

5.4.3 Comparison for the PCRB

This is where issues become confusing. The PCRB allows the tracks (the true tracks, not the observed data, which has always been random) to be random processes. There are two obvious methods for generating the tracks and comparing the bound to the sample MSE.

Method A

1. Generate one track set according to the constant velocity model, incorporating process noise.
2. Compute the PCRB for this track set using the process noise statistics.
3. Generate N versions of this track set by adding N realizations of the observation noise.
4. Run a practical tracker on each of these N sets of observed data.
5. Find the sample MSE.

Method B

1. Generate a track set according to the constant velocity model *without* incorporating process noise.
2. Compute the PCRB for this track set using the process noise statistics.
3. Generate N versions of this track set with N realizations of the process noise and N realizations of the observation noise.
4. Run a practical tracker on each of these N sets of observed data.
5. Find the sample MSE.

The difference between methods A and B lies in how the process noise is incorporated into the system. Method A is analogous the steps for computing the CRLB, except that the tracks are allowed to deviate from the constant velocity model. Method A seems to be the one most commonly used [7, 21, 37] and certainly gives useful results.

We feel, however, that Method B is technically the correct approach to comparing the sample MSE to the PCRB. Recall that the PCRB depends on the density

$$p(\mathbf{Z}_k, \mathbf{X}_k) = p(\mathbf{x}_0) \prod_{i=1}^{k} p(\mathbf{z}_i|\mathbf{x}_i) \prod_{j=1}^{k} p(\mathbf{x}_j|\mathbf{x}_{j-1}). \tag{165}$$

The process noise affects the $p(\mathbf{x}_j|\mathbf{x}_{j-1})$ term by determining the probability of the state changing from $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$. Imagine the N Monte Carlo trials in the sample MSE calculation

at time $k-1$. If we use Method A, the Monte Carlo trials will all have the same transition from $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$, and consequently, the trials are not samples of this distribution. On the other hand, using Method B, the Monte Carlo trials will provide a sampling from this distribution. For the bound computation, computing the bound around the true constant-velocity track has, in some sense, computed the bound around the "average" track for the set of initial conditions and particular constant-velocity model.

Although Method A seems popular, it does not appear to match the PCRB. Many researchers treat Method A as though it was computing some sort of *conditional* PCRB (conditioned on a specific set of tracks); such a conjecture does not necessarily follow from [38]. We suspect that many authors are, in essence, inadvertently treating a realization of the process noise as a "known" $\mathbf{u}_k$ (see (147)).

Method B does have its disadvantages, however. We are particulary interested in the bounds near target crossings, which is the interesting aspect of the MTT problem. However, when running Monte Carlo trials using Method B, the location and time of target crossings will change from run-to-run because of the different process noise realizations. Consequently, any trends in the MSE from target crossing will be averaged out. This is not an issue with Method A since all the trials have the same process noise realization and the targets will always cross paths at the same time. Figure 19 gives a visual demonstration of these issues. The scenario used for this demonstration is given in Figure 18.

Figure 19(a) shows the approximate[8] PCRBs and sample (Monte Carlo) MSEs obtained using Method A. Four scenarios were derived from the constant velocity model by adding four different realizations of the process noise. The four scenarios each include a target crossing near the middle of the scenario, but at distinct times. The effect of the target crossing is clearly visible both in the MSEs and in the PCRBs. According to the PCRB, some scenarios have a more difficult crossing than others, and this result also generally appears in the MSE. The "steady-state" levels of the PCRBs and the MSEs differ significantly. This goes back to the explanation for why we believe Method B is technically more sound

[8]The approximate PCRB is found using an EKF and the SME sum-of-products. The bound is approximate since the observation noise after the SME transformation is modelled as Gaussian, but is not actually Gaussian.

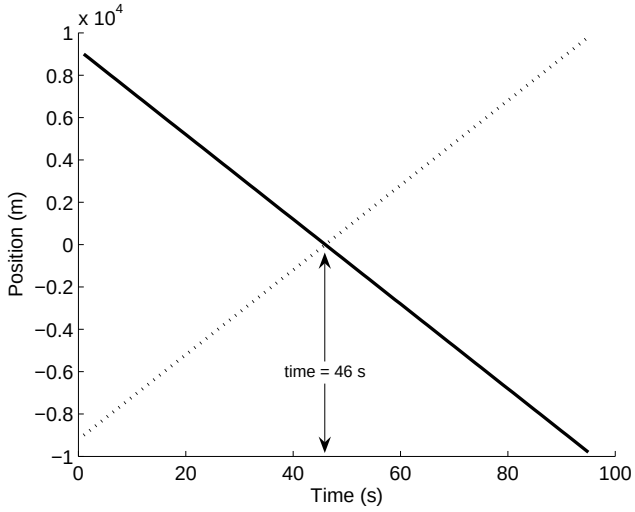
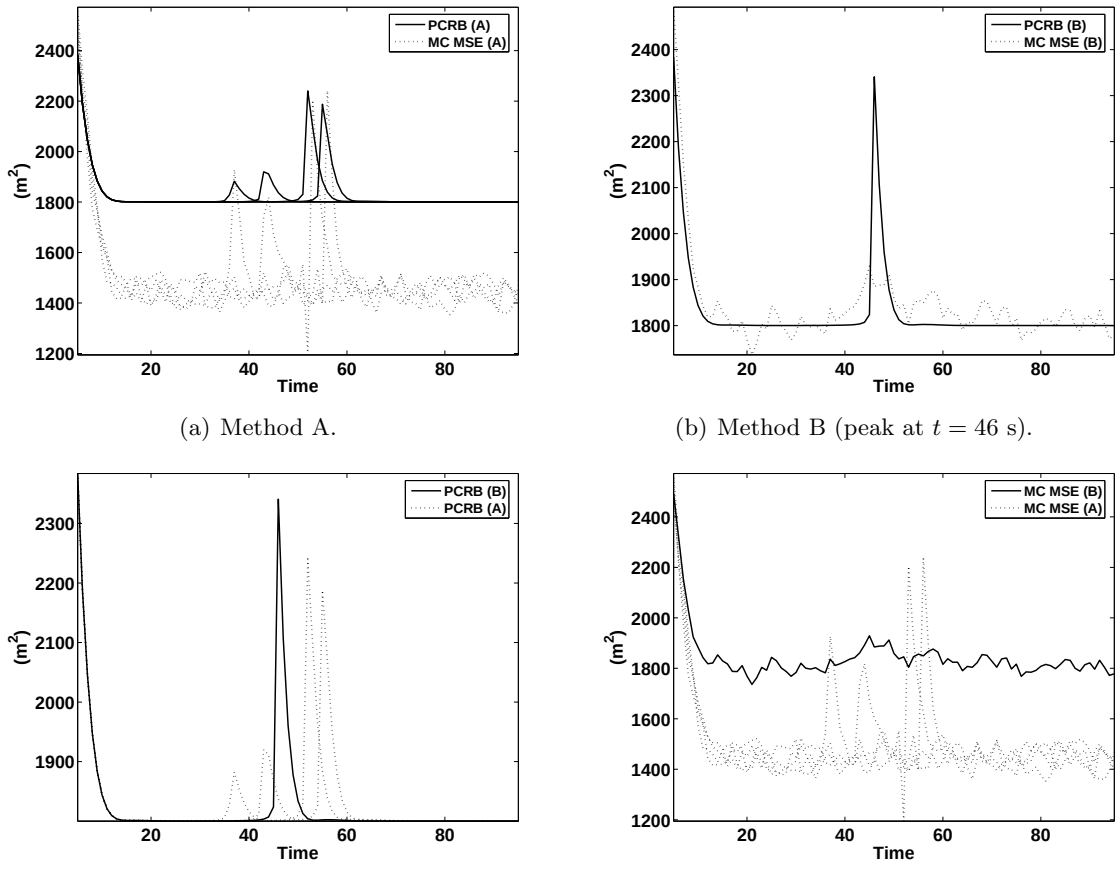


Figure 18: Example target paths in one dimension.



(a) Method A.

(b) Method B (peak at $t = 46$ s).

(c) PCRB.

(d) Monte Carlo MSE ($N = 3000$ trials).

Figure 19: Different methods for computing the PCRB and sample MSE.

than Method A with regards to [38]. The bounds are computed assuming that the state transitions at every time step are subject to a random process (process noise). The MSE in Method A, however, is computed for a fixed realization of the process noise.

Figure 19(b) shows the approximate PCRBs and sample (Monte Carlo) MSEs using Method B. In this case, there is only one bound - the bound calculated around the target trajectory without process noise, which represents an *average* trajectory. The tracks for the N Monte Carlo trials are generated with N different process and observation noise realizations. With this method, the "steady-state" levels of the PCRB and MSE match well. This is particularly fortuitous since the Monte Carlo error being computed is a "set error," and the PCRB theorems do not explicitly take this "set" nature into account. While the effect of the target crossing is clear in the PCRB, its effect is barely visible as small bump near the target crossing in the MSEs.

5.5 Bound versus Performance Prediction

All the CRLB and PCRB techniques used in this work are approximate. To efficiently compute the CRLB using Taylor's EKF method, we must assume a Gaussian observation noise, but the SME observation noise is not Gaussian. To efficiently compute the PCRB, we make the Gaussian assumption and also linearize the SME observation function. In spite of these approximations, computing the PCRB and MSE according to Methods A and B both provide useful results. Method A is good for capturing tracker behavior near target crossings, and Method B is good for capturing the steady-state MSE behavior. It is, however, clear from Figure 19 that neither PCRB method is actually lower bounding the MSE. For these reasons, we have decided to call the approximate PCRB methods performance prediction methods instead of lower bounds.

For the rest of this chapter, the performance predictions will be computed as an approximate PCRB around the true constant-velocity track (Method B), unless otherwise noted. For the sample MSE, we run tracking filters on this same constant-velocity track corrupted by observation noise. This is similar to Method A, just with the particular process noise

realization of all zeros. We chose this method to ensure that the effects from target crossings line up with the target crossings in the performance prediction. In some places, we have also computed the MSE according to Method B to verify the steady-state level of the performance prediction. In such places the MSE will be denoted as MSE-B. Before beginning to experiment with these performance prediction methods, we first briefly explore one case where it is reasonable to suppose our performance predictions are computing an approximate CRLB.

5.6 Approximate CRLB in One Dimension

Taylor's original formulation for computing a bound requires additive white Gaussian observation noise and zero process noise. As mentioned earlier, these restrictions do not generally apply to multiple target tracking scenarios. However, for reference, we can generate an associated[9] Kalman filter-based bound that does meet these requirements. We can also feed the EKF/SME-based and UKF/SME-based performance predictions zero process noise, which removes one of the two violations of the Taylor-style EKF/UKF bounding technique. We add the suffix "based" to emphasize that the bounds are meant to apply to any tracking algorithm, and not specifically to an SME tracker; the bounds just happen to be computed using SME. The non-Gaussian observation noise is still present in the EKF/SME-based and UKF/SME-based performance predictions (see Section 5.2.2).

Figure 20 shows an example of target paths in one dimension, and Figure 21 compares four items for this scenario: the MSE of an associated Kalman filter, the CRLB calculated with the associated Kalman filter, the EKF/Products-based performance prediction, and the UKF/Powers-based performance prediction. Each metric is computed as the variance/MSE on one dimension of one target. In Figure 21, the associated Kalman filter-based bound and the two SME-based performance predictions are indistinguishable.

Figure 22 shows a zoomed-in view of these three plots. We can see that the two SME-based performance predictions are slightly higher than the associated Kalman filter-based

[9]The associated Kalman filter is "clairvoyant," i.e., it knows the correct data association and runs a standard Kalman filter for each track.

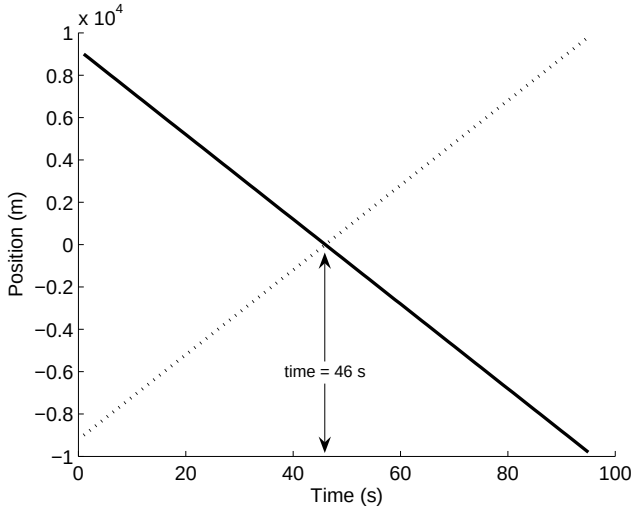


Figure 20: Example target paths in one dimension.

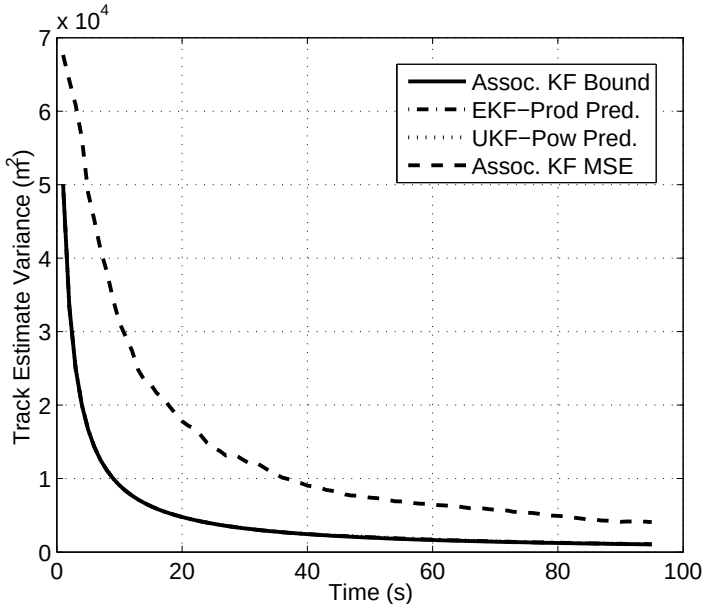


Figure 21: Comparison of three bounds and Monte Carlo results when process noise is set to zero in both the Monte Carlo runs and the bound computations.

bound. It is reassuring that the associated Kalman filter lower bound curve falls below the associated Kalman filter Monte Carlo MSE curve. We can consider the associated Kalman filter-based bound to be a true Cramér-Rao lower bound since it satisfies all the necessary assumptions; hence, something would be wrong if the MSE of the Monte Carlo trials fell below the bound. In this specific scenario, the SME performance predictions violate the Gaussian observation noise assumption, but not the prohibition on process noise.

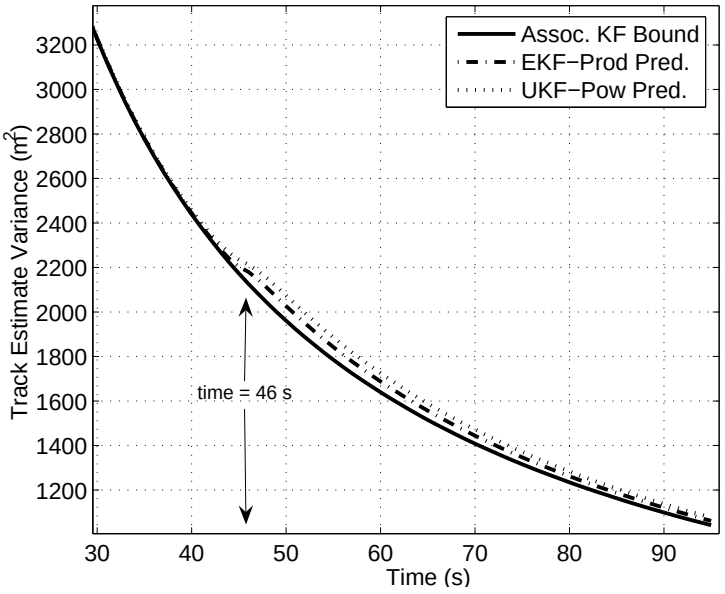


Figure 22: Comparison of three bounds when process noise is set to zero; zoom view.

Consequently, the nearly identical behavior of the associated Kalman filter-based bound and the two SME performance predictions suggests that the non-Gaussianity of the observation noise may not have a major impact on the two SME-based performance predictions. Also, note that there appears to be only a slight abberation in the bound/performance predictions at the target crossing near $t = 46$ s. Upon reflection, this intuitively makes sense; the difference between the associated Kalman filter and the SME is likely to be negligible when the targets are far apart and only come into play when they become close. Interestingly the effect of the crossing seems to retain its influence on the bound through the rest of the scenario; it is as if the bound remains aware that a target switch might have taken place.

Looking at the cross-covariance term provides another interesting insight. Figure 23 shows this term for the three performance predictions above. For the associated bound, it is automatically zero since independent Kalman filters are run for each target. However, for the SME filters, the crossing of the two targets is clearly visible as a valley in the cross-covariance curves. Despite no obvious affects from the target crossing in Figure 21, the filter is aware of the crossing via the cross-covariance terms.

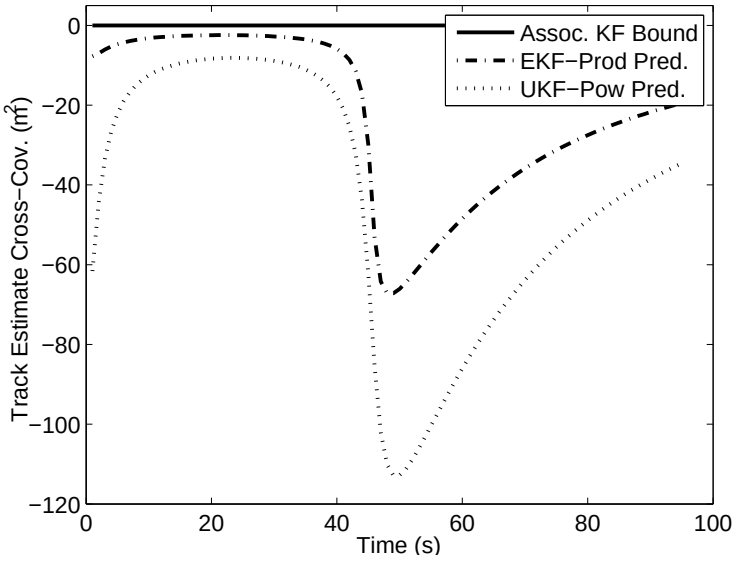


Figure 23: Cross-covariance bound term when process noise is set to zero.

5.6.1 Negative Cross Covariance

Aside from the clear indication of the target crossing in Figure 23, there is another distinguishing characteristic: the cross-covariance is negative. This result is not immediately intuitive. To understand this behavior, recall that the covariance of interest, assuming an unbiased estimator, is described by

$$P = E\begin{bmatrix} (x_1 - \hat{x}_1)^2 & (x_1 - \hat{x}_1)(x_2 - \hat{x}_2) \\ (x_1 - \hat{x}_1)(x_2 - \hat{x}_2) & (x_2 - \hat{x}_2)^2 \end{bmatrix}, \tag{166}$$

where x_1 and x_2 indicate the true positions and $\hat{x}_1$ and $\hat{x}_2$ indicate the estimated positions. Excluding the expectation, nothing inherent in this equation suggests the cross-covariance terms should be negative; indeed, in some cases, they will be positive. However, Figure 23 suggests the average behavior of the cross terms is negative. Suppose the targets are approaching each other, as in Figure 24. The estimated tracks have intentionally been drawn on the "inside" of the approaching targets. For a global nearest neighbor assignment algorithm, it is not hard to imagine that, in general, most data association errors will occur when both measurements lie in the region between the approaching targets. We saw in Chapter 2 that the SME-based particle filter using Gaussian likelihood approximations also

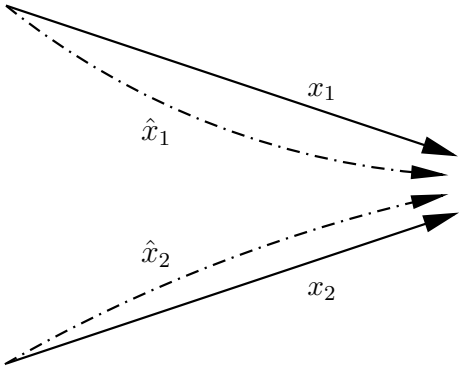


Figure 24: Illustration of approaching targets.

biases measurements toward each other. When the estimated tracks make an error in this way, the $(x_1 - \hat{x}_1)$ and $(x_2 - \hat{x}_2)$ will have opposite signs, and thus encouraging a negative cross-covariance.

5.7 Performance Predictions in One Dimension

5.7.1 Constant Velocity Targets

The previous section showed that the bounds computed without process noise exhibit no significant behavior at target crossings in the diagonal terms (although the cross terms showed some interesting behavior). While it provided useful analysis of the Gaussian observation noise assumption, the no-process-noise assumption is unrealistic. Hence, we now move to our PCRB-based performance predictions. Figure 25 shows the EKF/Sum-of-Products-based performance prediction for three different process noise levels[10] and a constant observation noise level. Figure 26 shows the same plot for the UKF/Sum-of-Powers-based predictions. The target paths are the same as in Figure 20. We again emphasize that "EKF/Sum-of-Products" and "UKF/Sum-of-Powers" refer to methods for computing bounds, and do not in any way imply that we are seeking bounds specifically on the performance of an EKF/Sum-of-Products or UKF/Sum-of-Powers based tracking filter.

There are several features to examine in these graphs. First, the steady-state performance prediction shows higher MSE values as the process noise rises. This result matches

[10]The process noise levels refer to the variance the filters are told to assume for each target, but no process noise is present in the paths. The targets are assumed to move independently.

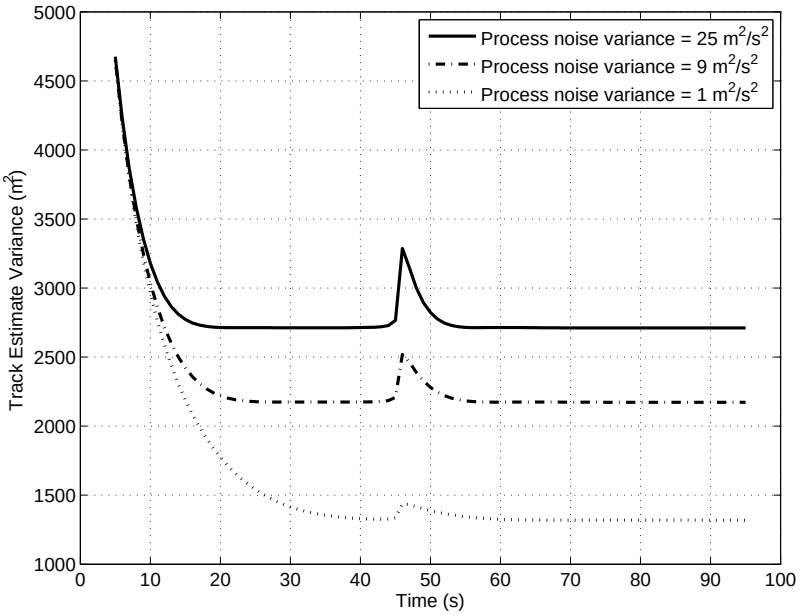


Figure 25: Effect of process noise on performance prediction, EKF/Sum-of-Products. The targets cross at $t = 46$ s.

with our intuition that an increase in process noise increases uncertainty, as the filter has difficulty locking onto a target. Next, notice the peak that appears near the target crossing at time 46 s. The peaks become more pronounced as the process noise rises. In particular, the peak is higher than just the steady-state change in process noise. This can be seen in Figure 27, which shows the result of dividing the higher curve, $var = 25$ m^2/s^2, by the lower curves, $var = 9$ m^2/s^2 and $var = 1$ m^2/s^2.

Figure 28 compares an EKF-SME performance prediction, a UKF-SME performance prediction, and Monte Carlo trials of the error covariance generated by a global nearest neighbor (GNN) tracker. In this scenario, the process noise is relatively high[11] ($var = 25$ m^2/s^2), and the observation noise is relatively low ($var = 1000$ m^2). There are a couple of features worthy of mention. First, the predictor does a good job of showing the peak that occurs near the target crossing for the GNN tracker. Figure 29 shows the same comparison, but with relatively low process noise ($var = 1$ m^2/s^2) and relatively high observation noise ($var = 5000$ m^2). Figure 30 is Figure 29 re-scaled to have the same axes as Figure 28.

[11]There is no process noise in the tracks used for the Monte Carlo runs, but the filter expects there to be. Hence, we would expect the performance predictions to be pessimistic, so the relationships in Figures 28 and 29 come as no surprise.

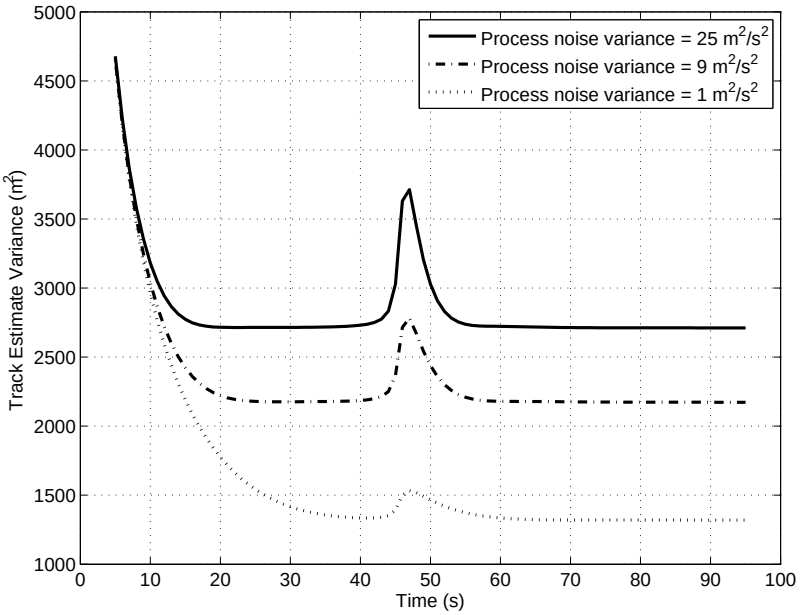


Figure 26: Effect of process noise on performance prediction, UKF/Sum-of-Powers. The targets cross at $t = 46$ s.

Again, the performance prediction matches the Monte Carlo trials well. The peak in error has a longer tail in both the prediction and the GNN results. The steady-state level for the predictions and the GNN has risen compared to the previous experiment. Also, the EKF-based and UKF-based performance predictions agree well with each other.

Figures 31 and 32 show the cross-covariance term for these two scenarios. The trends that appeared in the cross-covariance terms of the CRLB are also present in our performance prediction. The predicted cross-covariance is never positive and shows a large dip at the target crossing (time = 46 s). The predictor also matches well with the Monte Carlo MSE obtained using the GNN filter. An interesting observation from these plots is that the filter clearly responds to the target crossings by changes in the cross-covariance terms of the covariance matrix.

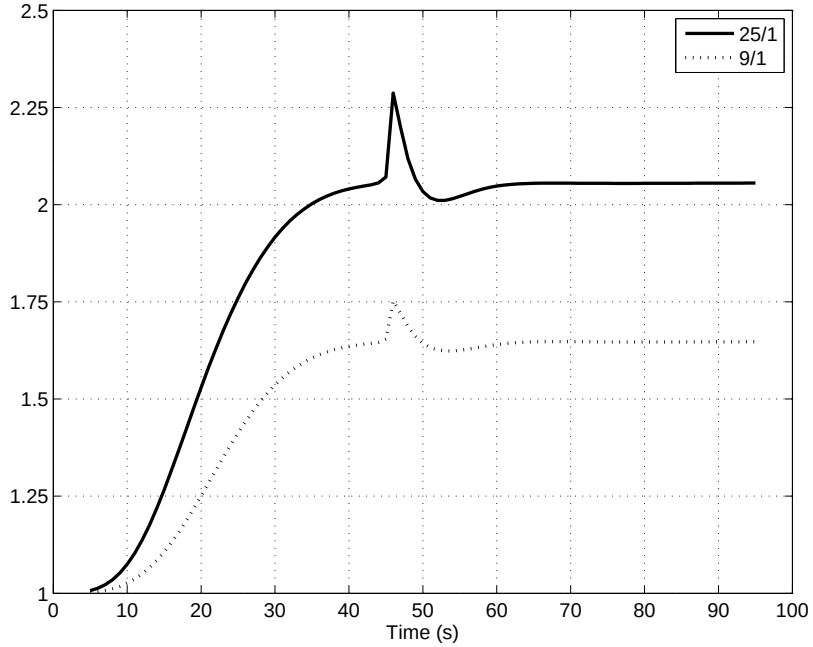


Figure 27: Examining peaks caused by process noise, EKF/Sum-of-Products. The targets cross at $t = 46$ s.

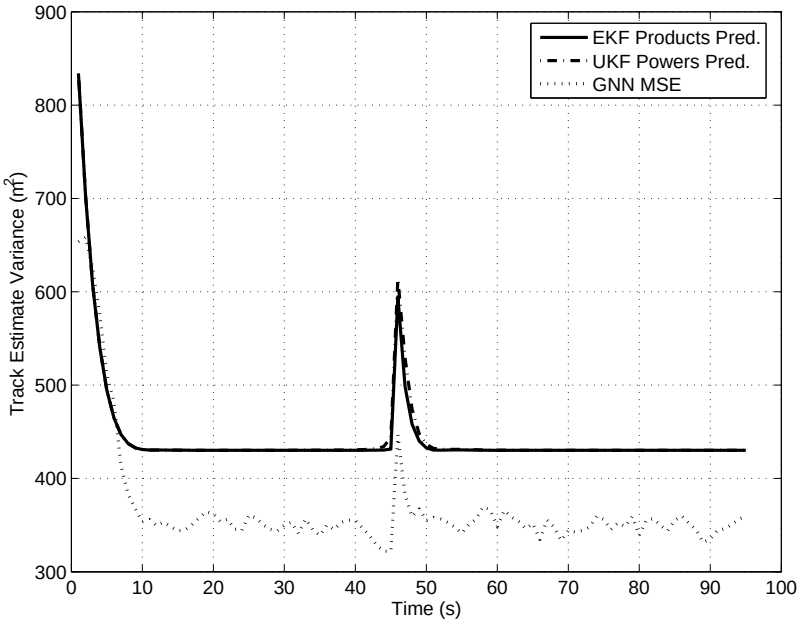


Figure 28: High process noise ($var = 25$ m^2/s^2), low observation noise ($var = 1000$ m^2).

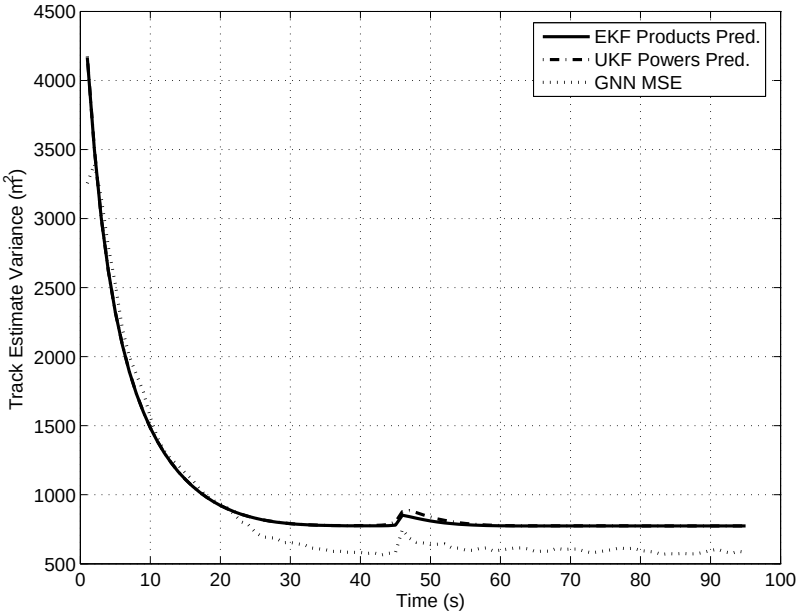


Figure 29: Low process noise ($var = 1\ \mathrm{m}^2/\mathrm{s}^2$), high observation noise ($var = 5000\ \mathrm{m}^2$).

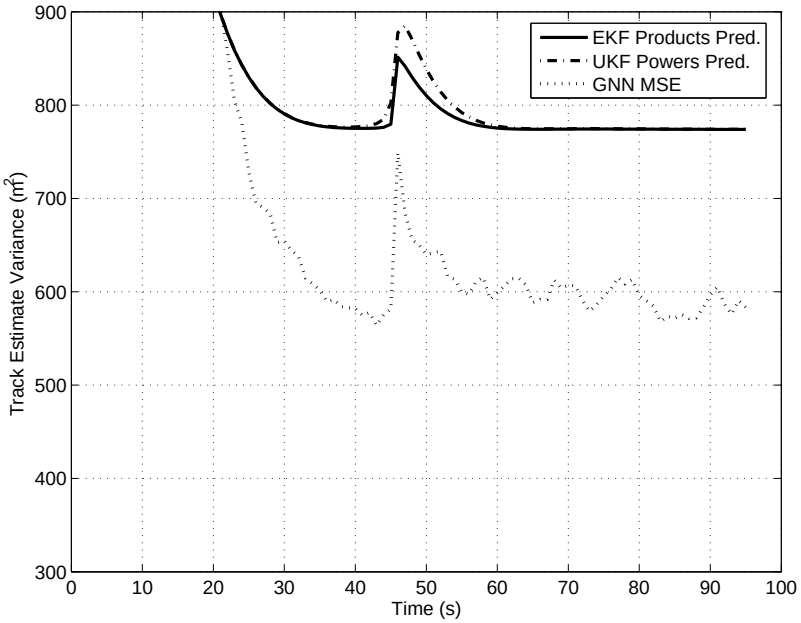


Figure 30: Low process noise, high observation noise - Re-scaled to match Figure 28.

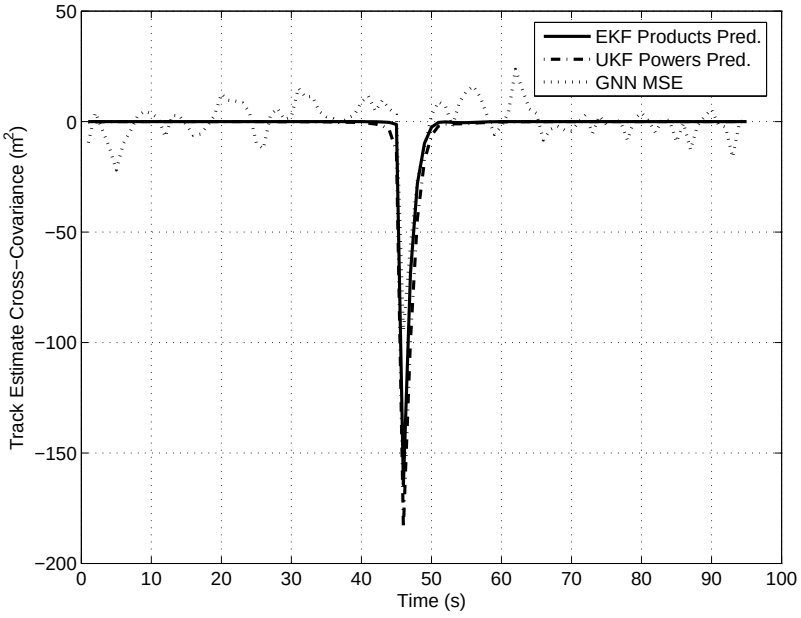


Figure 31: Cross covariance with high process noise ($var = 25$ m^2/s^2), low observation noise ($var = 1000$ m^2).

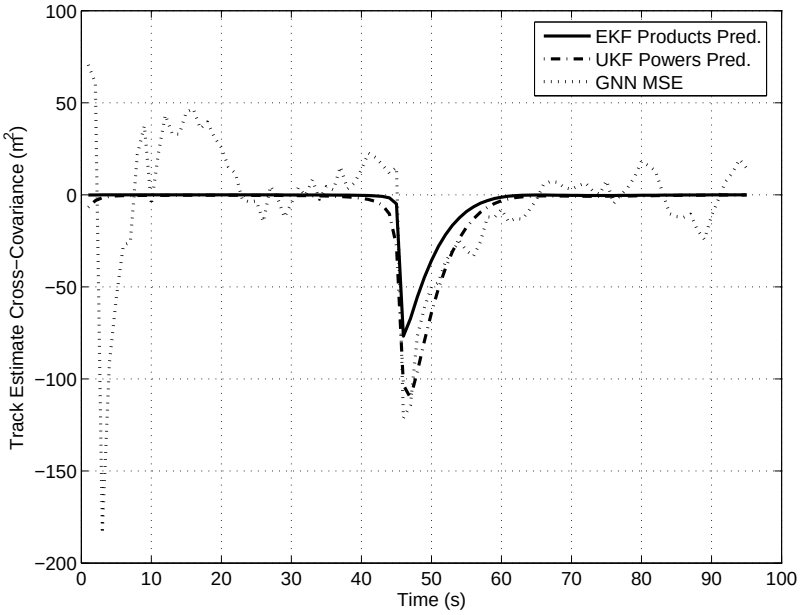


Figure 32: Cross covariance with low process noise ($var = 1$ m^2/s^2), high observation noise ($var = 5000$ m^2).

5.7.2 Maneuvering Targets

We are also interested in how the performance prediction works for scenarios with maneuvering targets. Figure 33 shows one such one-dimensional scenario. The targets in this scenario are moving directly towards each other (since they are in one dimension) at a constant velocity. Recognizing the impending collision, they decelerate to a stop and then accelerate away from each other.

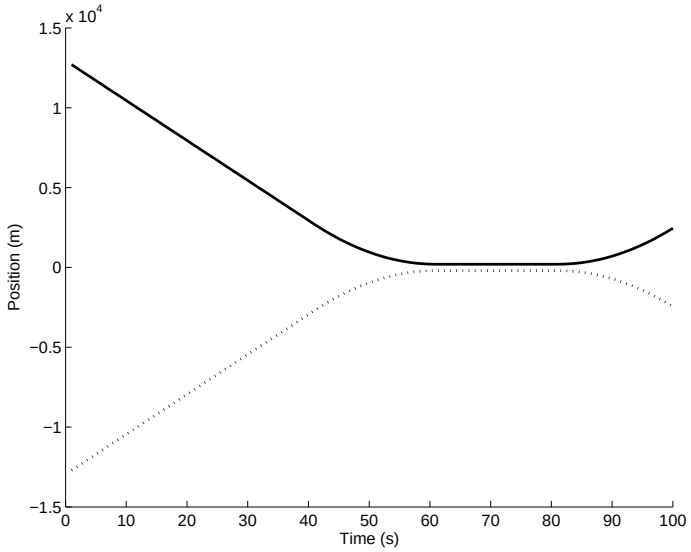


Figure 33: Paths for maneuvering targets.

To account for the significant deviation from the constant velocity assumption, the process noise needs to be modified. One simple approach is to scale the white process noise from the constant velocity assumption by a factor q, where q is a tuning parameter [5]. Our performance predictions can help choose this tuning parameter. We choose q to be $q = \sqrt{ka_m}$, where k is scaling constant and a_m is the maximum acceleration in the scenario.[12] We performed experiments with $k = 1, 3, 5, 7, 10$ and compared the averaged set MSE to the EKF/Products-based bound. Figure 34 shows the result of this experiment.

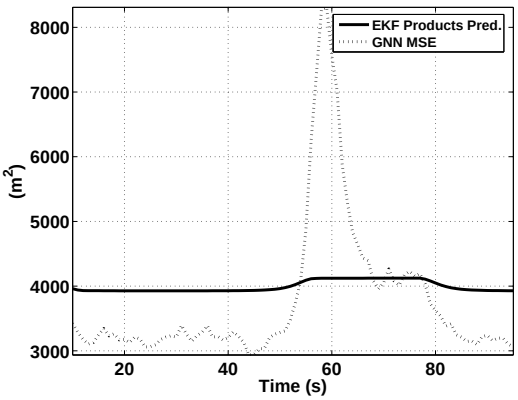
Figure 34 gives a good feel for the appropriate choice of k. When k is set too low, the filter is unable to track through the first deceleration. This can be seen from the MSE spike in Figure 34(a). On the other hand, when k is set too high, the filter essentially uses

[12]This is a slight abuse of a heuristic suggested in [5], but it should be sufficient for our purposes.

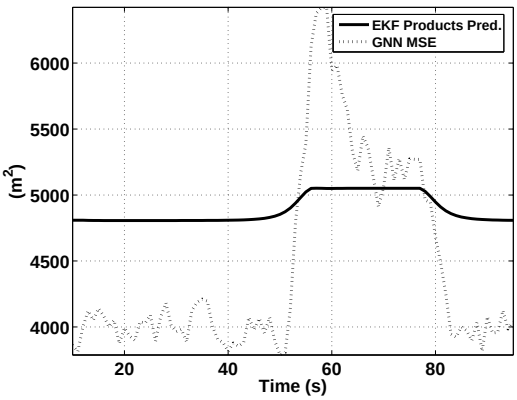
the observations as the estimate, which can be seen from the relatively high MSE floor in Figure 34(e). So a good choice for k is the lowest value where the MSE mimics the prediction (notice the overshoot for $k = 1$ is too high to be considered "mimicking"). Figure 34(f) shows the set MSE for the GNN estimator for all five values of k. The values $k = 3$ and $k = 5$ provide a good tradeoff between the height of the MSE spike at the first maneuver and the floor MSE level. The rest of this section will proceed with $k = 5$.

With the process noise covariance chosen, another experiment we performed was to change the closest approach distance of the two targets. Presumably, the closer the targets approach each other, the more the tracking problem becomes. Figures 35 and 36 show the performance predictions for different approach distances for the EKF/Products-based prediction and the UKF/Powers-based prediction, respectively. Only the region of interest is shown in the figures. The results match our intuition - the smaller the separation distance, the larger the potential error. The UKF/Powers-based method generates a rather high prediction for the 400 m separation. In fact, raising k to higher levels causes the UKF/Powers-based prediction to break down completely. The shapes of the two curves are also of interest. The EKF/Products-based prediction exhibits a tiny overshoot at the end of the deceleration (at time ≈ 57 s) similar to, although much smaller, than the overshoot demonstrated by the GNN (see Figure 37). The UKF/Powers-based prediction, on the other hand, displays another trend altogether. In short, the UKF/Powers-based prediction seems inapplicable in this scenario.

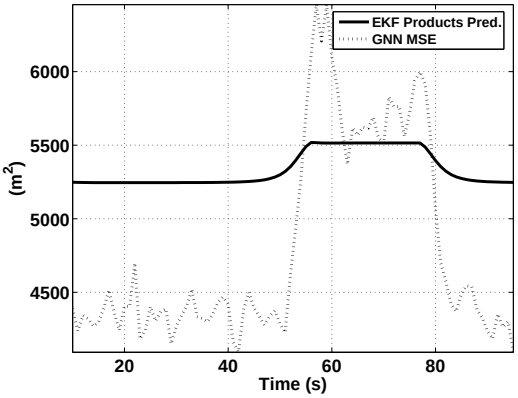
The last two figures for this experiment compare the MSE from a GNN tracker to the performance bounds. Figure 37 and Figure 38 show the results for a minimum separation of 400 m and 800 m, respectively . The performance prediction captures the GNN MSE peak well when the targets have a minimum separation distance of 400 m. However, the GNN tracker does not have a peak in the MSE when the minimum separation distance is 800 m. This can be attributed to the discrete nature of the GNN approach - it typically either finds the correct association, or it does not. If the separation is sufficient, the GNN filter will maintain the track. If the separation is not sufficient, the GNN filter usually will either make its error when the targets first get close together, or not make the error at all.



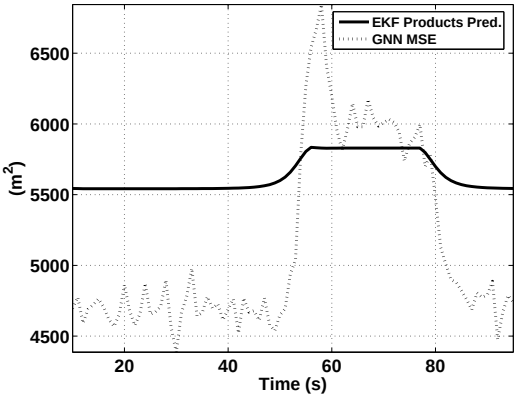
(a) $k = 1$.



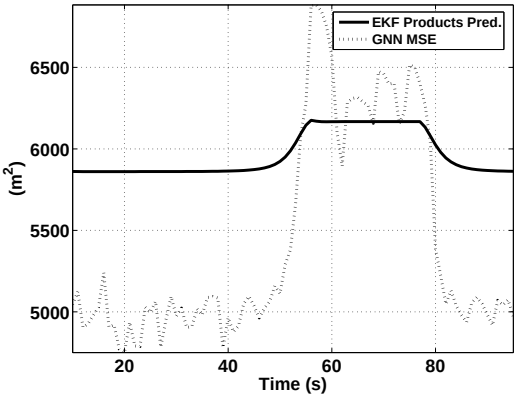
(b) $k = 3$.



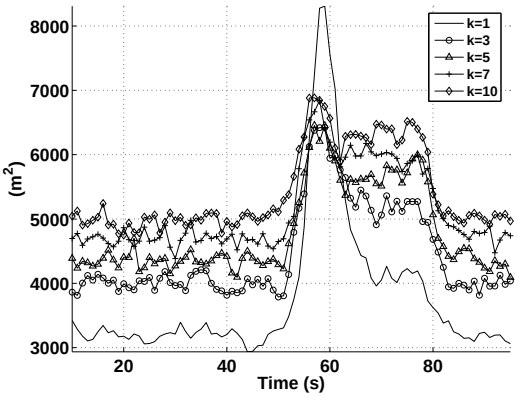
(c) $k = 5$.



(d) $k = 7$.



(e) $k = 10$.



(f) All GNN MSE Results.

Figure 34: Performance predictions for choosing a process noise scaling parameter.

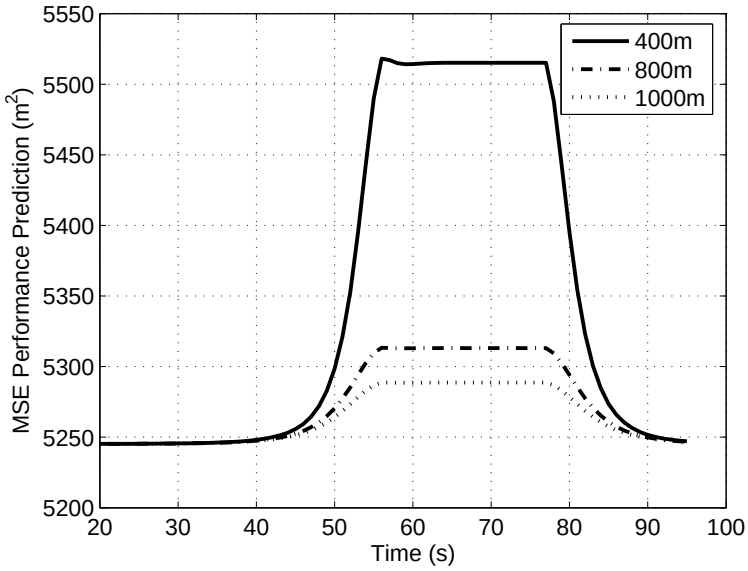


Figure 35: EKF/Products-based performance prediction for maneuvering targets. The legend indicates the minimum separation distance.

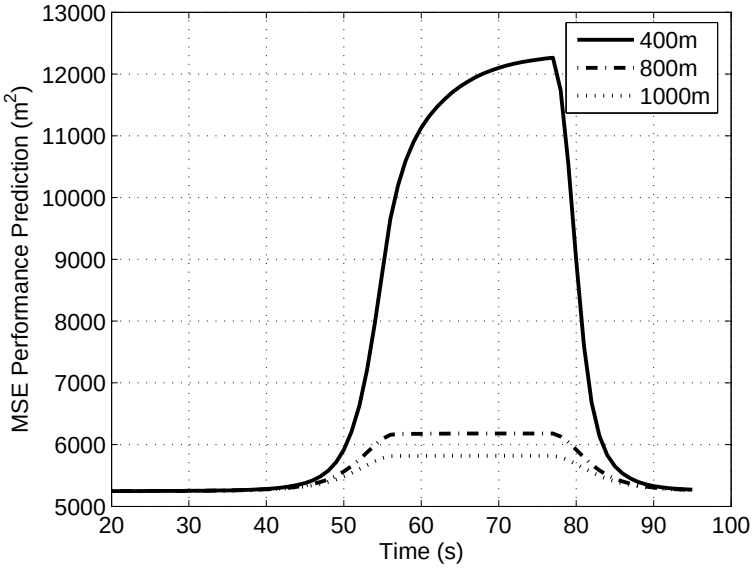


Figure 36: UKF/Powers-based performance prediction for maneuvering targets. The legend indicates the minimum separation distance.

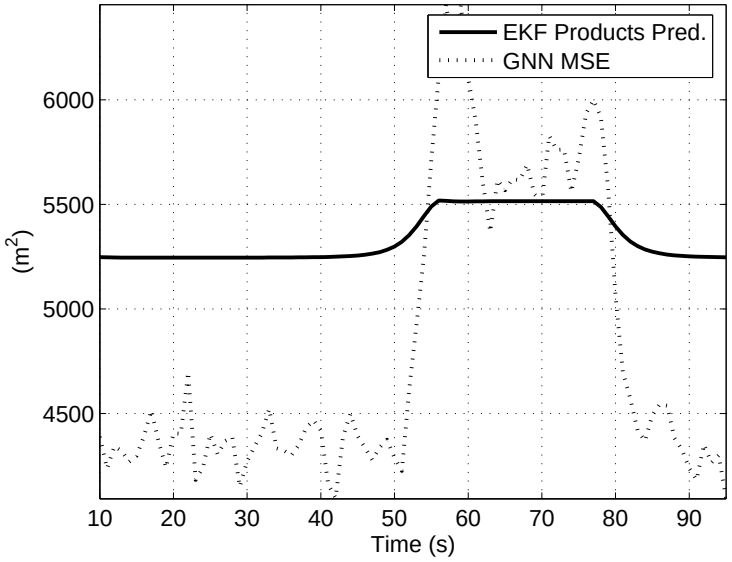


Figure 37: EKF/Products-based performance predictions and GNN MSE for maneuvering targets; minimum separation 400 m.

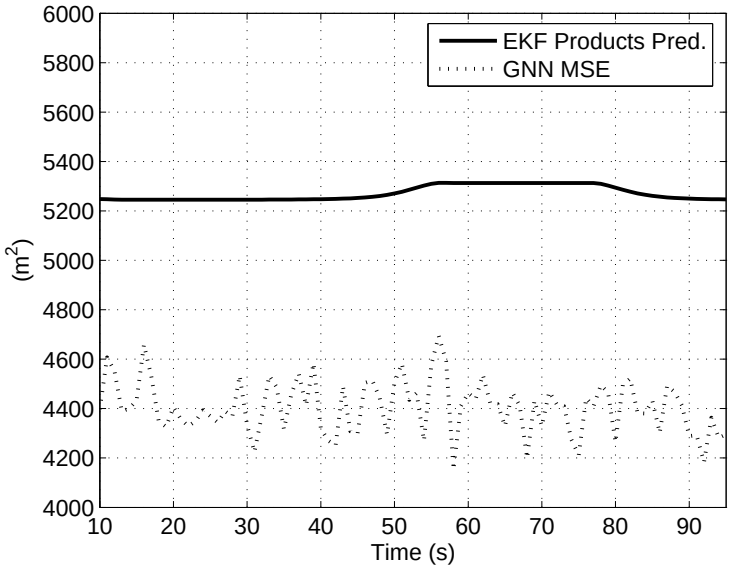
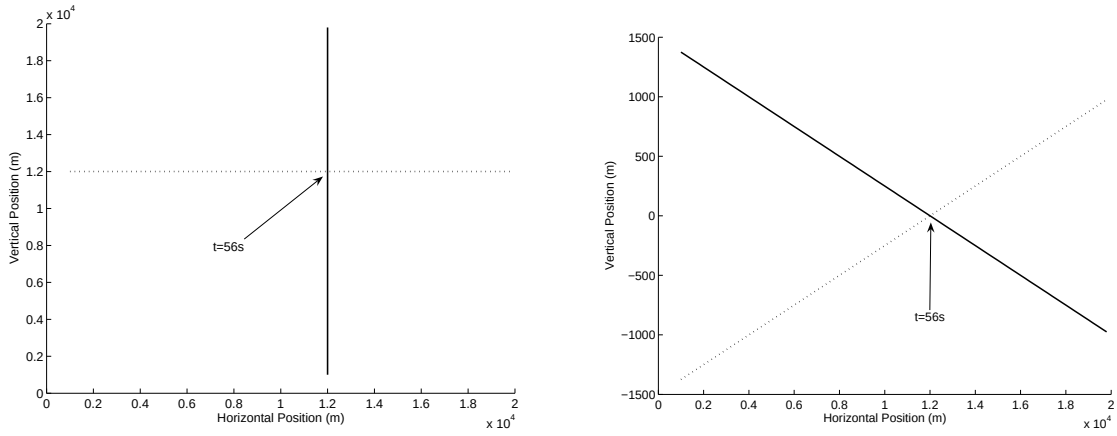


Figure 38: EKF/Products-based performance predictions and GNN MSE for maneuvering targets; minimum separation 800 m.

5.8 Performance Predictions in Two Dimensions

The next series of experiments deals with computing performance predictions for targets moving in two dimensions. The two-dimensional SME implementation follows that given in Chapter 3. Figure 39(a) shows the first scenario, in which the two targets start on the axes with one moving horizontally and the other vertically and both targets moving with the same speed. This guarantees that the targets will cross paths at the same time. Figure 39(b) shows a second scenario. In this scenario, the targets start at the left edge and move right. Their horizontal velocities are the same and their vertical velocities are the opposites of each other, again to guarantee crossing through the same point at the same time. Scenario 2 was designed to be more difficult than Scenario 1, since the approach angle in Scenario 2 is smaller. For both scenarios, the observation and process noise variances in each dimension were set to 1×10^4 m^2 and 1 m^2/s^2, respectively. The values were chosen to match the values used in Chapter 3.



(a) Scenario 1. (b) Scenario 2.

Figure 39: Two-dimensional motion scenarios.

Since, in general, the EKF/Products-based performance prediction produced better results, we have focused on that implementation here. We compare the performance prediction to the average MSE of Monte Carlo trials of an EKF/Products-based SME filter (for a fixed track set), global nearest neighbor (GNN) filter (for a fixed track set), and a GNN MSE run on "Method B" tracks (i.e., many random tracks in addition to noisy data). Figure 40 shows the performance prediction and MSE results for Scenario 1. The results agree

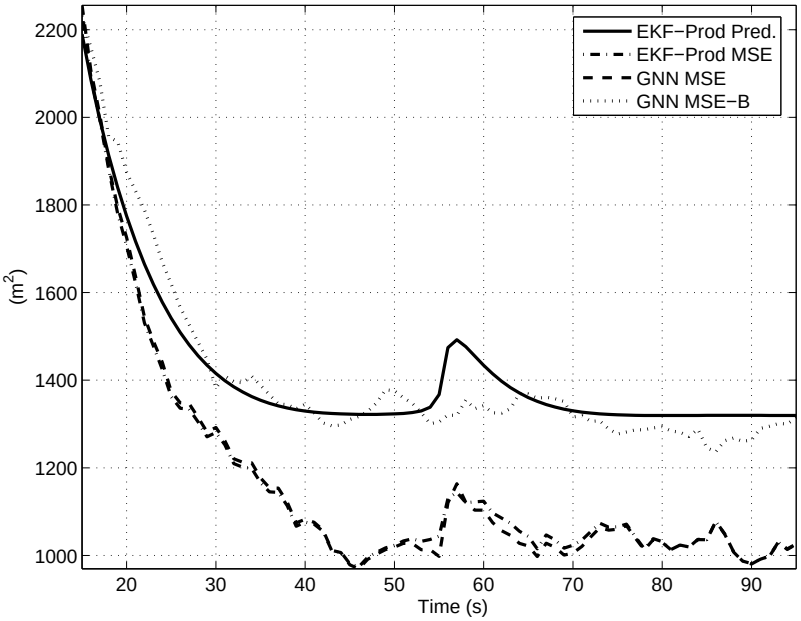


Figure 40: Performance prediction compared to filter MSE, Scenario 1. The targets cross at $t = 56$ s.

with our intuition that this should be a relatively easy scenario since the targets are moving perpendicularly. The bump at the target crossing is small in both the prediction and the fixed-track-set MSE metrics. From the MSE-B metric, the performance prediction is doing a good job of estimating the steady-state MSE. There is no bump seen in the MSE-B line since crossings occur in different places for different track realizations, and hence average out over many runs.

Figure 41 shows the EKF/Products-based[13] prediction versus the average MSE from Monte Carlo trials of two trackers for Scenario 2. The large peak in the performance prediction and the trackers' MSE at the target crossing match our intuition that this scenario is more difficult than the previous one. Again, the GNN MSE-B metric shows that the performance prediction estimates the steady-state level well. One interesting aspect of the GNN MSE plot is the sharp dip at time 56 s. This dip is a result of the targets crossing precisely at time 56 s. The dip occurs because the filter is usually tracking the targets well enough to estimate their positions as being nearly the same at this time. It also points out

[13]The reader should not read too much into the fact that we happen to be using an EKF/Products formulation for both the performance prediction and one of the trackers.

that the peak in the performance prediction and in the MSE occurs slightly after the target crossing. This makes sense if you consider confusion as increasing the longer the targets are in the same vicinity.

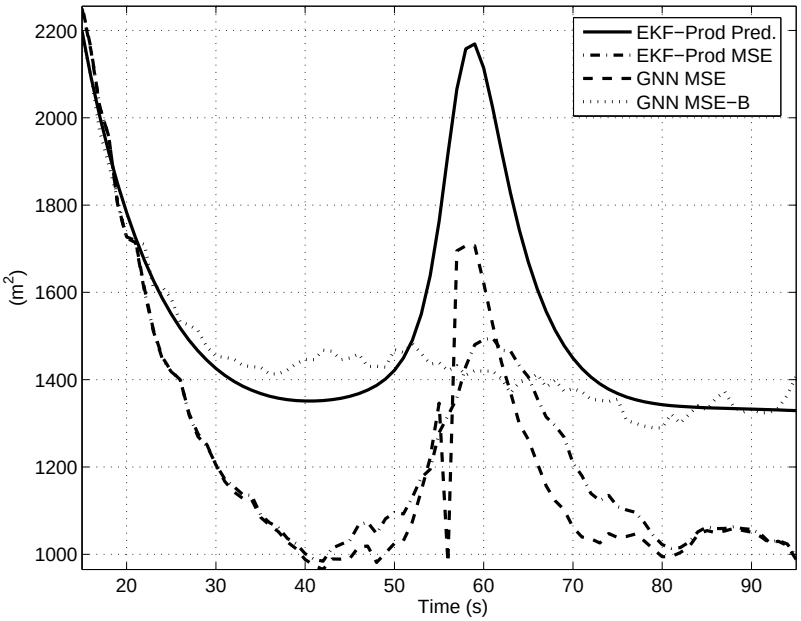


Figure 41: Performance prediction compared to filter MSE, Scenario 2. The targets cross at $t = 56$ s.

5.9 Performance Predictions in Three Dimensions

The final series of experiments gives a couple of brief examples of the performance prediction for targets moving in three dimensions. The prediction uses the quaternion-based SME described in Section 4. Again, we examine two scenarios: an easier one and a more difficult one. The first (easier) scenario is the same as that shown in Figure 15 in Chapter 4.3. The observation and process noise variances in each dimension were set to 1×10^4 m^2 and 10 m^2/s^2, respectively. The values were chosen to match the values used in Chapter 4. For the GNN approach, having three dimensions actually makes the data association problem easier because the observations have to be closely spaced in all three dimensions to be confusing. The increase in process noise over the two-dimensional case helps compensate for this effect and make the tracking problem more challenging.

Figure 42 compares the EKF/Products-based and the EKF/Powers-based predictions to

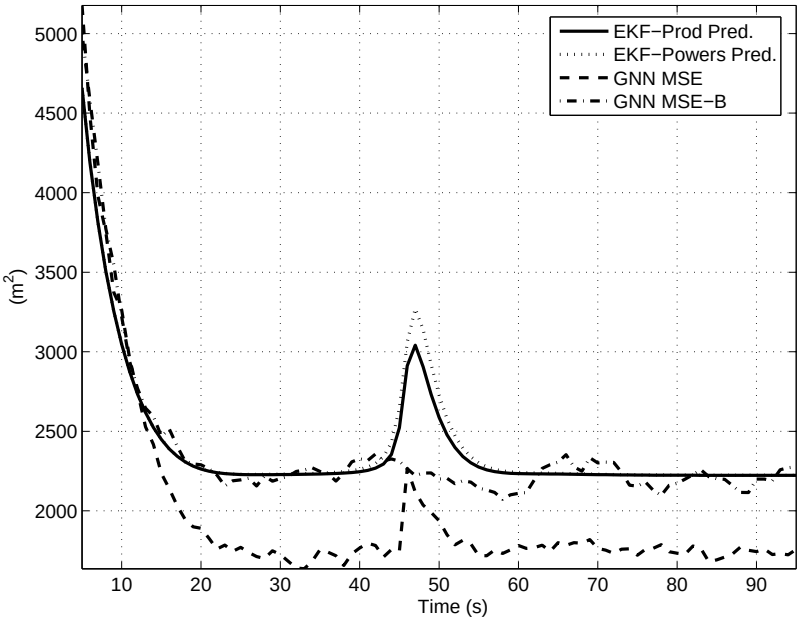


Figure 42: Performance predictions and averaged GNN filter MSE, Scenario 1. The targets cross at $t = 46$ s.

the GNN tracker sample MSE (for a fixed track set) and MSE-B (averaged over many track sets) for Scenario 1. The small peak when the targets are closely spaced suggests that this is a relatively easy scenario. The two performance predictors are nearly identical and show the same behavior as the GNN sample MSE plot. The steady-state level of the predictors closely matches the MSE-B plot. There is no bump seen in the MSE-B line since crossings occur in different places for different track realizations, and hence average out over many runs.

The second (harder) scenario is the same as that shown in Figure 16. The observation and process noise variances in each dimension were set to 1×10^4 m^2 and 10 m^2/s^2, respectively. Figure 43 compares the EKF/Products-based and the EKF/Powers-based predictions to the GNN tracker sample MSE and sample MSE-B. In this scenario, the performance prediction has a quite large bump near the target crossing, which indicates a difficult scenario. The GNN MSE also has a bump at the target crossing, but it is not nearly as large as the bump in the performance prediction. In this case, the prediction seems to overstate the difficulty to some degree. The MSE-B plot is included to show that the performance prediction accurately predicts the steady-state MSE.

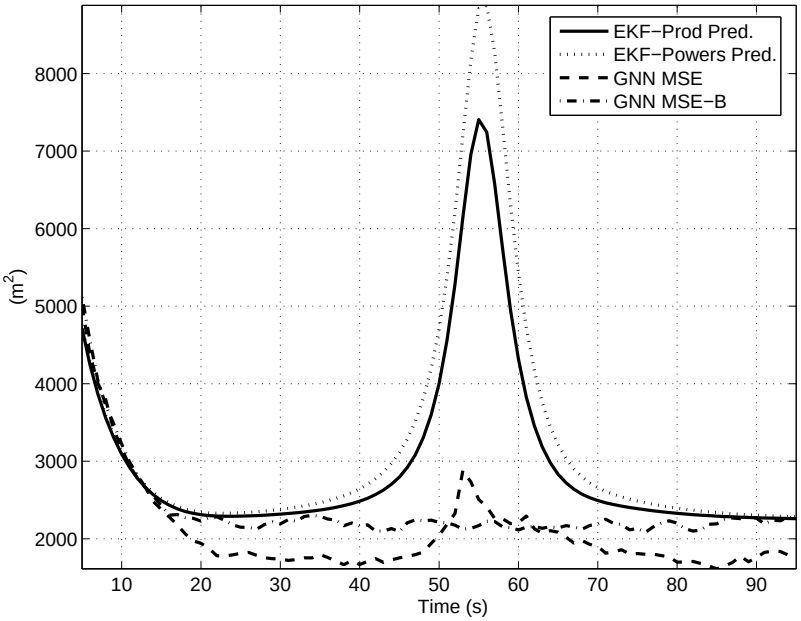


Figure 43: Performance predictions and averaged GNN filter MSE, Scenario 2. The targets cross at $t = 46$ s.

5.10 Performance Predictions for System Design

The previous sections studied the use of performance predictions from a system analysis viewpoint. Provided a set of target and tracking system characteristics, the performance prediction may be used to predict how the system will behave. A different perspective is to use the performance predictions for system design rather than analysis. In this situation, the desired tracker performance, rather than the system characteristics, is specified, and the question is how to achieve this performance. In our case, we assume that performance is measured in terms of tracker mean squared error computed using Method B, as discussed earlier. We also assume that the observation noise of our sensors and the observation interval are the two parameters we can vary, with the realization that lower observation noise may correspond to a more expensive sensor, and operational issues may bound the operation interval.

Imagine that the targets' constant velocity motion is described by Figure 44. Our first study, shown in Figure 45, examines the relationship between target process noise and predicted MSE. We have assumed that the performance prediction has two interesting values: the value at the target crossing and the steady-state value. We have kept the observation

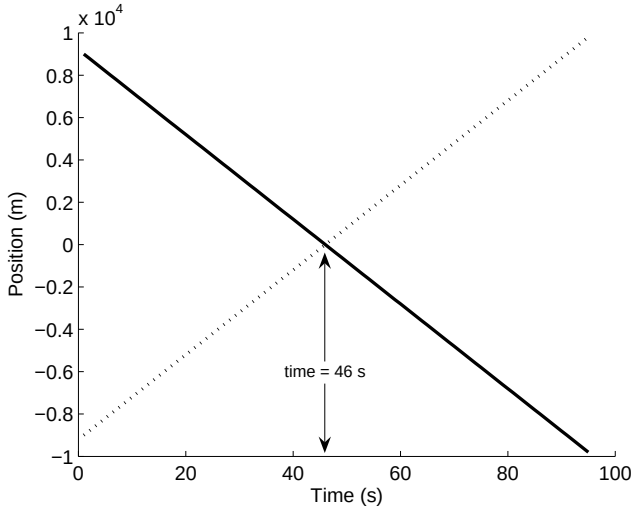


Figure 44: Example target paths in one dimension.

noise variance constant at 1000 m^2. Results from 3000 Monte Carlo trials (Method B) using the global nearest neighbor filter (GNN) are included as well to demonstrate the validity of the performance prediction results. From the figure, we can see how the MSE varies with changing process noises.

Figure 45 is useful for predicting how a specific tracker would perform against targets exhibiting different levels of process noise. However, a system designer does not have control over the maneuverability, i.e. the process noise, of the targets, but he may have control over the frequency of target observations. To explore this relationship, we have assumed that the process noise is discretized as follows. We assume that the continuous-time process noise for a single target moving in one dimension is distributed as a white zero-mean Gaussian process with power σ_w^2. The discretized process noise is then distributed according to $\mathcal{N}(0, Q)$, where T is the time between observations and

$$Q = \sigma_w^2 \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^2 & T \end{bmatrix}. \tag{167}$$

See [2] for the derivation of this discretization.

Figure 46 shows a plot of observation interval versus predicted or sample MSE. The value at the target crossing and the steady-state value are both plotted. For the GNN

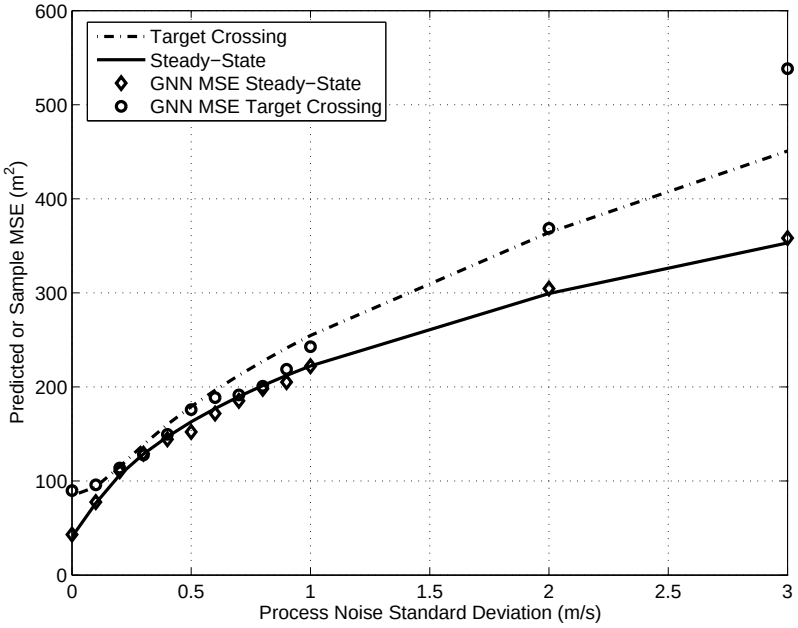


Figure 45: Example of process noise analysis.

filter, the sample MSE is calculated from 3000 Monte Carlo trials using Method B. The observation noise variance is fixed at 1000 m^2, and the continuous-time process noise power is fixed at 1 m^2/s^2. In the figure, we see close agreement between the sample MSE from the GNN filter and the performance prediction. From a system design point of view, the graph also allows the system designer to see how performance can be improved by decreasing the observation interval. However, such gains may be offset by the realization that many real sensors have performance that decreases as the observation interval decreases.

As suggested in the previous paragraph, another key aspect of a tracking system is the sensor observation noise. The same experiment that is shown in Figure 45 for process noise can be run with observation noise. The results of this experiment are shown in Figure 47. The continuous-time process noise power was fixed at 0.1 m^2/s^2 to minimize its influence, and the observation interval was 1 s. In the figure, we see that the sample MSE computed from the GNN Monte Carlo trials matches the prediction quite well. We also see that the difference between the error at the target crossings and the steady-state error grows rapidly with increasing observation noise.

The last set of figures is useful for analyzing performance trade-offs between reducing

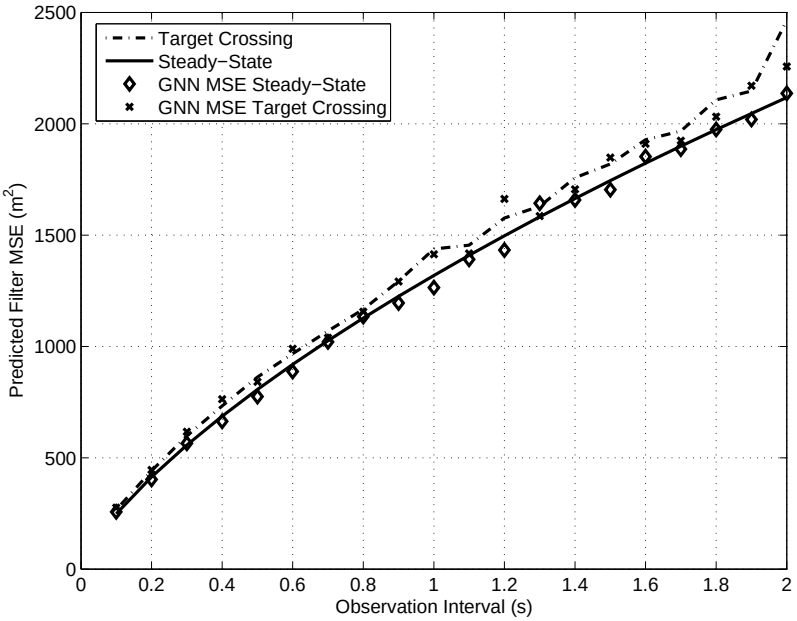


Figure 46: Example of observation interval analysis.

sensor observation noise and reducing the sensor observation interval (although, as mentioned before, these parameters are often linked given a particular sensor). Figures 48 and 49 show curves of observation noise versus predicted MSE for the steady-state and peak values, respectively. In each figure, this curve has been calculated for the observation intervals listed along the right-hand side of the figure. For this example, the continuous-time process noise power has been fixed at 1 m^2/s^2. To illustrate how these plots might be useful in system design, suppose that a steady-state MSE of 2000 m^2 is desired. Many options are apparent from Figure 48. On one end, we can observe the targets every 0.5 s and with an observation noise standard deviation of approximately 182 m. On the other end, we can observe the target every 1.9 s, but our observation noise standard deviation must then decrease to 98.5 m. Of course, the sampling interval could be increased even further with a corresponding decrease in observation noise. Essentially, less frequent observations result in fewer data points; hence, to achieve the same performance, each observation must be less noisy to compensate for the loss of data.

One limitation of the design approach described here is that the analysis depends on the targets paths. In all the previous plots, we used the scenario shown in Figure 44, although

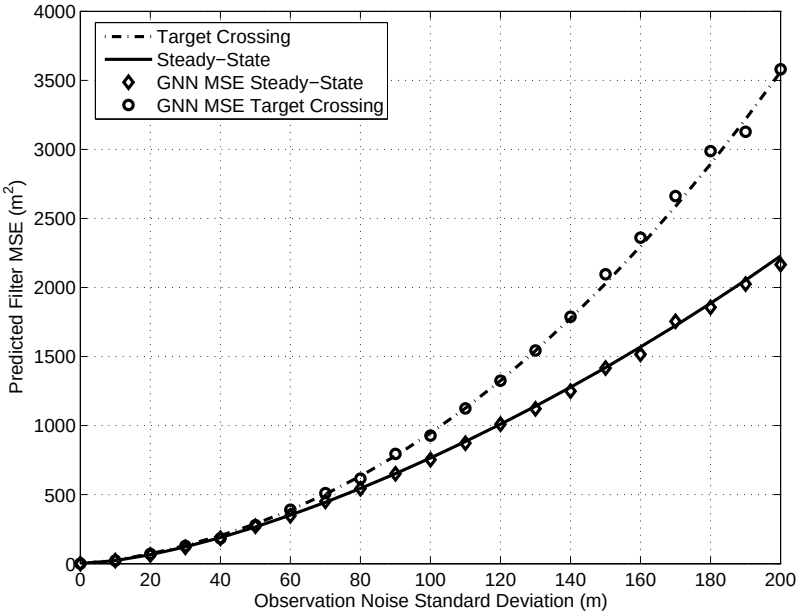


Figure 47: Observation noise analysis.

there is nothing unique about the scenario. The analysis can be performed for scenarios with many targets moving in one, two, or three dimensions. Observation noise and interval are the two parameters we have considered at this time, but extending the analysis to include other motion models such as a constant acceleration model would be straightforward.

5.11 Conclusions

This chapter presented a performance prediction approach based on the posterior Cramér-Rao lower bound for some multiple target tracking problems. While the performance prediction is not a strict bound, the technique presented does provide a means for quick and efficient comparative studies between different MTT tracking scenarios. Although the *absolute* level of the predictions rarely match Monte Carlo findings, the overall *trends* are clearly visible. For constant velocity (with or without process noise) targets, the performance prediction has been shown to mimic the behavior of GNN and SME multiple target tracking algorithms. For maneuvering targets, the prediction has been shown to be helpful in the analysis of tuning a process noise parameter. Another significant contribution of this chapter has been to clarify the meaning of the PCRB for target-tracking problems. Finally, we have demonstrated how to use our performance predictions as a system design tool.

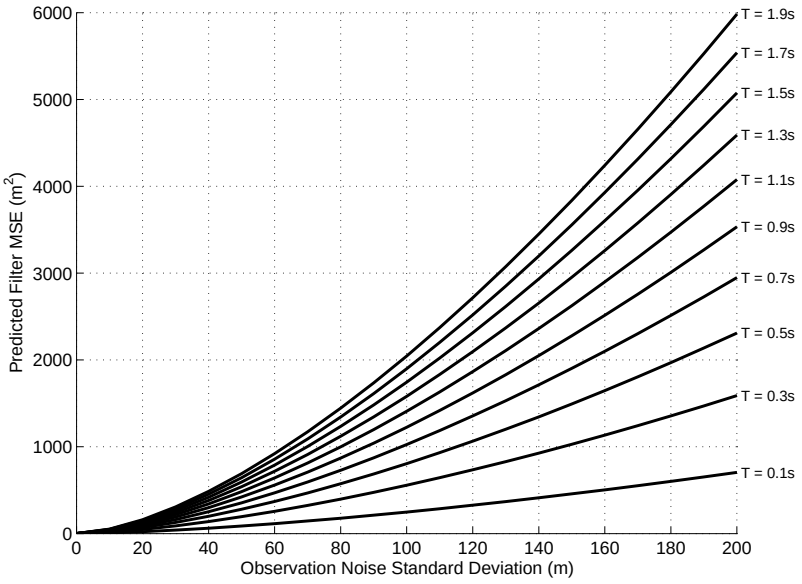


Figure 48: Steady-state performance predictions for a range of observation intervals.

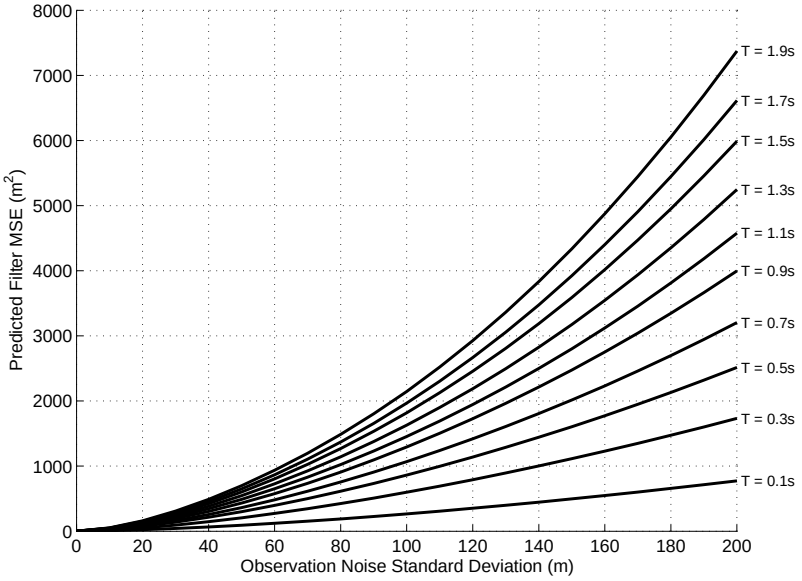


Figure 49: Performance predictions at target crossings for a range of observation intervals.

CHAPTER VI

CONCLUSIONS

The goal of this work was to develop an efficient method of computing performance predictions for multiple target tracking problems. The key idea was to use the symmetric-measurement-equations (SME) approach to convert the MTT problem into a nonlinear filtering problem. Before computing the performance predictions with the SME, however, we began by adding techniques to the body of knowledge on SME-based tracking.

We started our investigation into SME-based tracking by exploring several new nonlinear filtering techniques for one-dimensional SME-based tracking. After the original early work on the SME concluded [18–20, 24, 35], the unscented Kalman filter [16, 17, 41, 43] was introduced as a potentially powerful, but equally efficient, replacement for the extended Kalman filter. Chapter 2 demonstrated, much to our surprise, that the UKF cannot handle the nonlinearity posed by the most previously studied SME, the sum-of-products. However, we also demonstrated that the UKF paired with the sum-of-powers SME, an SME form previously thought to be inferior, outperformed any implementation of one-dimensional SME-based tracking with the EKF. In Chapter 2, we also experimented with particle filters, but our results showed that a straightforward particle filter implementation would not be sufficient to handle the SME nonlinearities if Gaussian likelihood approximations are used.

Motivated by the success of the sum-of-powers/UKF combination in the one-dimensional case, Chapter 3 developed a new sum-of-powers SME for tracking targets moving in two dimensions. Our results show that the UKF could not handle this new two-dimensional sum-of-powers SME or the existing two-dimensional sum-of-products SME due to problems with covariances becoming singular after the SME transformation when target estimates become closely spaced. When implemented with an EKF, we showed that the performance of the sum-of-products and sum-of-powers forms were similar, with the sum-of-products showing a slight advantage. The troubles with the UKF were disappointing since the UKF

is far easier to implement than the EKF.

Chapter 4 introduced a novel extension of the SME approach to the case of tracking targets in three dimensions. While two other methods exist, both have at least one significant drawback. One method cobbles together a solution by combining a one-dimensional SME and a two-dimensional SME, but this requires targets to be consistently well-resolved in one dimension. A second method uses polynomials to extend the SME to N-dimensions, but this requires an unnecessarily large set of symmetric measurement equations. Our method has neither of these drawbacks and has sufficient performance for our main goal of computing MTT performance predictions.

Chapter 5 presented the main contribution of this work: an efficient method of predicting performance for multiple target tracking problems. The method allows a user to specify the process and observation noises of the tracking system as well as the paths targets follow. With this setup, our techniques predict overall trends in the mean squared error that a practical algorithm could be expected to demonstrate. Enabled by the SME approaches described in Chapters 2-4, targets can move in one, two, or three dimensions. Our method is based on the posterior Cramér-Rao lower bound. Some misunderstandings exist in the literature about the proper application of this bound in target tracking scenarios. We remedied this confusion in Chapter 5.

Besides the contribution of an efficient MTT performance prediction method, there are also several negative conclusions that resulted from this work. Aside from the special case of one-dimensional tracking, the UKF could not handle the SME nonlinearities studied, although the EKF worked decently in the same situations. This runs contrary to the mystique surrounding the UKF. The Taylor series analysis in Chapter 2 demonstrates that the performance of the SME approach, at least when implemented with Kalman filters, may degrade significantly as the number of targets increases. Finally, the SME approach is extremely difficult to work with, and we caution anyone else about pursuing it unless they have new ideas about how to make working with it less palatable. In particular, a fix for the singularity troubles of the UKF would help researchers avoid the tedious and error-prone derivations needed to implement the EKF.

There are several other potential avenues for pursuing further work in this area. A nonlinear filtering methodology that handles the SME nonlinearities well would be a major contribution. The main limitation of all the filters implemented in this dissertation is that they all approximate the complicated SME likelihood densities with Gaussians. Alternative algorithms that allow more accurate likelihood models, such as, perhaps, Daum's "exact filters" could potentially yield vast improvements [8]. As far as computing performance predictions, missed detections and false alarms are significant aspects of MTT that have not yet been incorporated. Lee [24] looked into modifying SME formulations to handle these effects in target tracking; a logical next step would be to see if these techniques could be applied to performance predictions.

Other researchers have applied Levenberg-Marquardt algorithms [4] and neural networks [30] to SME tracking. It would be interesting to try those approaches to the experimental scenarios in this dissertation for comparison, and in particular to apply those techniques to our new two-dimensional sum-of-powers SME formulation and our new three-dimensional quaternion-based SME formulation.

In our two-dimensional and three-dimensional examples, we assumed that the sensors gave Cartesian coordinates corrupted by noise that was independent in the different coordinates and had a fixed variance. Real radar systems typically give measurements in spherical (for three dimensions) or cylindrical (for two dimensions) coordinates, and the measurements are independent in range and azimuth (and elevation, in the three-dimensional case). In [35], this was handled for a three-dimensional SME-based tracker by converting spherical measurements to Cartesian coordinates in a preprocessing stage, along with an associated transformation of the covariance in the spherical domain to a covariance in the Cartesian domain. The resulting transformed covariance is no longer diagonal, and would also depend on the target location. Although the resulting SME approximations needed to implement an EKF are still conceptually straightforward, they become far more tedious to compute and result in much more complicated expressions. These issues will need to be resolved to develop multiple target tracking performance predictions for realistic radars.

A popular alternative to converting non-Cartesian data to Cartesian data is to track

using the non-Cartesian data directly via a nonlinear filter such as the EKF. One possibility that has not yet been explored in SME research would be to build the nonlinear measurement mappings (i.e., Cartesian to spherical conversion) directly into the SME formulation. This would become necessary with some sensors (i.e., in bearings-only tracking) whose instantaneous measurements cannot be uniquely mapped into Cartesian space.

One underlying and unresolved question asks what other SME formulations are available. We have repeatedly shown that different nonlinear filters differ in terms of which SME formulations they perform well with. The sum-of-products and sum-of-powers are the most obvious SMEs to propose, but there are certainly others; for instance, one could imagine proposing a new set of SMEs by summing the sum-of-products and sum-of-powers polynomials, and it is not difficult to imagine other possibilities. Of course, the invertibility of any new proposed SME (up to a permutation) must be confirmed. It might be the case that the EKF, the UKF, or some other filter has an easier time with the resulting SME, even though the SME itself is more complicated. Similarly, there might be an advantage to having a larger number of equations than is technically necessary for invertibility of the SME up to permutation, in terms of compatibility with some filter classes.

All SME research to date has solely involved position measurements from a single sensor. One obvious extension would be to the case where velocity measurements are available as well. For instance, in two-dimensional tracking, if velocities in x and y are available in addition to positions, then we would have a measurement with four coordinates; in this case, quaternions might be appropriate, as they naturally can hold four elements. More realistically, we might have measurements of range, azimuth, and range-rate, but not an additional azimuth-rate measurement.

In theory, the SME formulation involves no loss of information, so its apparent lack of performance in our target tracking experiments is vexing. Our conjecture is that the Achilles' heel of all current SME-based tracking implementations, including the ones in this thesis, is the Gaussian approximation to the likelihood. More accurate approximations, and the creation of filters than can effectively exploit them, could bring SME-based tracking into the mainstream.

APPENDIX A

SUPPLEMENT TO CHAPTER II: GAUSSIAN APPROXIMATIONS FOR SME MEASUREMENT NOISE PROBABILITY DENSITIES

A.1 Sum of Products

Recall that for the sum-of-products SME, the measurements are found by taking the received data, y, and calculating a pseudomeasurement, $\tilde{y}_{prod}$, as follows:

$$\tilde{y}_{prod} = \begin{pmatrix} y_1 + y_2 + y_3 \\ y_1y_2 + y_1y_3 + y_2y_3 \\ y_1y_2y_3 \end{pmatrix}. \tag{168}$$

Also, recall that y_i can be written as $x_i + u_i$ where x_i is the truth and u_i is an additive Gaussian noise term with variance σ_{u_i}. Restrictions on the SME approach require that y_i can be written as

$$y_i = g(x_1, x_2, \cdots, x_N) + v_i \quad \text{for } i = 1 \cdots N. \tag{169}$$

In general, the distribution of v_i depends on $x_1, x_2, \cdots, x_N$. Following the derivation in [20], let R be the covariance matrix of the new measurement noise, v. For the three-target case, R can be approximated as

$$R \approx \sigma_1^2 V_1 V_1^T + \sigma_2^2 V_2 V_2^T + \sigma_3^2 V_3 V_3^T, \tag{170}$$

where

$$\sigma_1^2 = \sigma_{u_1}^2 + \sigma_{u_2}^2 + \sigma_{u_3}^2, \tag{171}$$

$$\sigma_2^2 = \sigma_{u_1}^2\sigma_{u_2}^2 + \sigma_{u_1}^2\sigma_{u_3}^2 + \sigma_{u_2}^2\sigma_{u_3}^2, \tag{172}$$

$$\sigma_3^2 = \sigma_{u_1}^2\sigma_{u_2}^2\sigma_{u_3}^2, \tag{173}$$

and

$$V_1 = \begin{pmatrix} 1 & 1 & 1 \\ x_2 + x_3 & x_1 + x_3 & x_1 + x_2 \\ x_2 x_3 & x_1 x_3 & x_1 x_2 \end{pmatrix}, \tag{174}$$

$$V_2 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ x_3 & x_2 & x_1 \end{pmatrix}, \tag{175}$$

$$V_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{176}$$

A.2 Sum of Powers

To simplify calculations in the sum-of-powers case, each measurement was assumed to have the same noise variance, such that $\sigma_{u_1}^2 = \sigma_{u_2}^2 = \sigma_{u_3}^2 = \sigma_u^2$. For the sum of powers,

$$y_{pow} = \begin{pmatrix} y_1 + y_2 + y_3 \\ y_1^2 + y_2^2 + y_3^2 - 3\sigma_u^2 \\ y_1^3 + y_2^3 + y_3^3 - 3\sigma_u^2(y_1 + y_2 + y_3) \end{pmatrix}, \tag{177}$$

where the σ_u^2 terms are included to ensure that the measurement noise after the sum-of-powers transformation is zero mean. The derivation required to approximate the new measurement noise covariance matrix, R, for the sum of powers is long and tedious, so the details have been omitted here. However, the result is the following:

$$R \approx \begin{pmatrix} 3\sigma_u^2 & 2\sigma_u^2 z_1 & 3\sigma_u^2 z_2 + 9\sigma_u^4 \\ 2\sigma_u^2 z_1 & 4\sigma_u^2 z_2 + 6\sigma_u^4 & 6\sigma_u^2 z_3 + 12\sigma_u^4 z_1 \\ 3\sigma_u^2 z_2 + 9\sigma_u^4 & 6\sigma_u^2 z_3 + 12\sigma_u^4 z_1 & 9\sigma_u^2 z_4 + 36\sigma_u^4 z_2 + 45\sigma_u^6 \end{pmatrix}, \tag{178}$$

where

$$z_1 = x_1 + x_2 + x_3, \tag{179}$$

$$z_2 = x_1^2 + x_2^2 + x_3^2, \tag{180}$$

$$z_3 = x_1^3 + x_2^3 + x_3^3, \tag{181}$$

$$z_4 = x_1^4 + x_2^4 + x_3^4. \tag{182}$$

Although (178) must have been used in [18], it does not seem to have explicitly appeared in the literature. We give the result here to save other researchers time in rederiving it.

APPENDIX B

SUPPLEMENT TO CHAPTER III: EKF IMPLEMENTATION OF THE TWO-DIMENSIONAL SUM-OF-POWERS SME

B.1 Gaussian Noise Approximation

Assume that the observations can be modelled as $(x_i + u_{x_i}, y_i + u_{y_i})$, where x_i, y_i are the truth and u_{x_i}, u_{y_i} are additive Gaussian noise terms with variances $\sigma^2_{x_i}$, $\sigma^2_{y_i}$. Restrictions on the SME approach require that z_i can be written as

$$z_i = g(x_1, x_2, \cdots, x_N, y_1, y_2, \cdots, y_N) + v_i \quad \text{for } i = 1 \cdots 2N \tag{183}$$

for an N-target system. The noise process, v_i should be zero mean and, in general, depends on $x_1, x_2, \cdots, x_N$. Following the derivation in [20], let R be the covariance matrix of the new measurement noise, v. Then,

$$R = E[v(n)v(k)^T], \tag{184}$$

where $v(n)$ is a $2N$x1 noise vector at time n.

To avoid tedious calculations, each measurement was assumed to have independent observation noise with the same variance such that $\sigma^2 = \sigma^2_{x_1} = \sigma^2_{x_2} = ... = \sigma^2_{x_N} = \sigma^2_{y_1} = \sigma^2_{y_2} = ... = \sigma^2_{y_N}$. For the two-target case, R can be approximated as

$$R \approx \sum_{i=1}^{3} V_i q_i q_i^T V_i^T, \tag{185}$$

where

$$q_1^T = [u_{x_1} \quad u_{y_1} \quad u_{x_2} \quad u_{y_2}], \tag{186}$$

$$q_2^T = [u^2_{x_1} \quad u^2_{y_1} \quad u^2_{x_2} \quad u^2_{y_2}], \tag{187}$$

$$q_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2(u_{x_1}u_{y_1} + u_{x_2}u_{y_2}) \end{pmatrix}, \tag{188}$$

and

$$V_1 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 2x_1 & -2y_1 & 2x_2 & -2y_2 \\ 2y_1 & 2x_1 & 2y_2 & 2x_2 \end{pmatrix}, \tag{189}$$

$$V_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{190}$$

$$V_3 = \mathrm{I}_4. \tag{191}$$

For the three-target case, R can be approximated as

$$R \approx \sum_{i=1}^{5} V_i q_i q_i^T V_i^T, \tag{192}$$

where

$$q_1^T = [u_{x_1} \quad u_{y_1} \quad u_{x_2} \quad u_{y_2} \quad u_{x_3} \quad u_{y_3}], \tag{193}$$

$$q_2^T = [u_{x_1}^2 \quad u_{y_1}^2 \quad u_{x_2}^2 \quad u_{y_2}^2 \quad u_{x_3}^2 \quad u_{y_3}^2], \tag{194}$$

$$q_3^T = [u_{x_1}^3 \quad u_{y_1}^3 \quad u_{x_2}^3 \quad u_{y_2}^3 \quad u_{x_3}^3 \quad u_{y_3}^3], \tag{195}$$

$$q_4^T = [u_{x_1}u_{y_1} \quad u_{x_2}u_{y_2} \quad u_{x_3}u_{y_3} \quad 0 \quad 0 \quad 0], \tag{196}$$

$$q_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -3(u_{x_1}u_{y_1}^2 + u_{x_2}u_{y_2}^2 + u_{x_3}u_{y_3}^2) - 2\sigma^2(x_1 + x_2 + x_3) \\ 3(u_{x_1}^2u_{y_1} + u_{x_2}^2u_{y_2} + u_{x_3}^2u_{y_3}) - 2\sigma^2(y_1 + y_2 + y_3) \end{pmatrix}, \tag{197}$$

and

$$V_1= \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 2x_1 & -2y_1 & 2x_2 & -2y_2 & 2x_3 & -2y_3 \\ 2y_1 & 2x_1 & 2y_2 & 2x_2 & 2y_3 & 2x_3 \\ 3(x_1^2-y_1^2) & -6x_1y_1 & 3(x_2^2-y_2^2) & -6x_2y_2 & 3(x_3^2-y_3^2) & -6x_3y_3 \\ 6x_1y_1 & 3(x_1^2-y_1^2) & 6x_2y_2 & 3(x_2^2-y_2^2) & 6x_3y_3 & 3(x_3^2-y_3^2) \end{pmatrix}, \quad (198)$$

$$V_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 3x_1 & -3x_1 & 3x_2 & -3x_2 & 3x_3 & -3x_3 \\ 3y_1 & -3y_1 & 3y_2 & -3y_2 & 3y_3 & -3y_3 \end{pmatrix}, \quad (199)$$

$$V_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & -1 \end{pmatrix}, \quad (200)$$

$$V_4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ -6y_1 & -6y_2 & -6y_3 & 0 & 0 & 0 \\ 6x_1 & 6x_2 & 6x_3 & 0 & 0 & 0 \end{pmatrix}, \quad (201)$$

$$V_5 = \mathrm{I}_6. \quad (202)$$

B.2 Jacobian Matrix

The Jacobian matrix, $\mathcal{J}$, for the two-dimensional, two- target sum-of-powers SME is given by

$$\mathcal{J} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 2x_1 & -2y_1 & 2x_2 & -2y_2 \\ 2y_1 & 2x_1 & 2y_2 & 2x_2 \end{pmatrix}, \tag{203}$$

and for the three-target case, it is given by

$$\mathcal{J} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 2x_1 & -2y_1 & 2x_2 & -2y_2 & 2x_3 & -2y_3 \\ 2y_1 & 2x_1 & 2y_2 & 2x_2 & 2y_3 & 2x_3 \\ 3(x_1^2 - y_1^2) - 2\sigma^2 & -6x_1y_1 & 3(x_2^2 - y_2^2) - 2\sigma^2 & -6x_2y_2 & 3(x_3^2 - y_3^2) - 2\sigma^2 & -6x_3y_3 \\ 6x_1y_1 & 3(x_1^2 - y_1^2) - 2\sigma^2 & 6x_2y_2 & 3(x_2^2 - y_2^2) - 2\sigma^2 & 6x_3y_3 & 3(x_3^2 - y_3^2) - 2\sigma^2 \end{pmatrix}. \tag{204}$$

APPENDIX C

SUPPLEMENT TO CHAPTER IV: EKF IMPLEMENTATION OF QUATERNION-BASED SME

C.1 Gaussian Noise Approximation

Assume that the observations can be modelled as $(x_i + u_{x_i}, y_i + u_{y_i}, z_i + u_{z_i})$, where x_i, y_i, z_i are the truth and u_{x_i}, u_{y_i}, u_{z_i} are additive Gaussian noise terms with variances $\sigma^2_{x_i}$, $\sigma^2_{y_i}$, $\sigma^2_{z_i}$. Restrictions on the SME approach require that m_i can be written as

$$m_i = g(x_1, x_2, \cdots, x_N, y_1, y_2, \cdots, y_N, z_1, z_2, \cdots, z_N) + v_i \quad \text{for } i = 1 \cdots 3N, \tag{205}$$

for an N-target system where $g(\cdot)$ represents a set of symmetric measurement equations. Following the derivation in [20], let R be the covariance matrix of the new measurement noise, v. Then,

$$R = E[v(n)v(k)^T], \tag{206}$$

where $v(n)$ is a $3N$x1 noise vector at time n.

To make the calculation convenient, each measurement was assumed to have independent observation noise with the same variance such that $\sigma^2 = \sigma^2_{x_1} = \sigma^2_{x_2} = ... = \sigma^2_{x_N} = \sigma^2_{y_1} = \sigma^2_{y_2} = ... = \sigma^2_{y_N} = \sigma^2_{z_1} = \sigma^2_{z_2} = ... = \sigma^2_{z_N}$.

C.2 Two-Target Sum-of-Products

For the two-target case, R_{prod} can be approximated as

$$R_{prod} \approx \sum_{i=1}^{2} V_i q_i q_i^T V_i^T, \tag{207}$$

where

$$q_1^T = [u_{x_1} \quad u_{y_1} \quad u_{z_1} \quad u_{x_2} \quad u_{y_2} \quad u_{z_2}], \tag{208}$$

$$q_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ u_{x_1}u_{x_2} - u_{y_1}u_{y_2} - u_{z_1}u_{z_2} \\ u_{x_1}u_{y_2} + u_{x_2}u_{y_1} \\ u_{x_1}u_{z_2} + u_{x_2}u_{z_1}) \end{pmatrix}, \tag{209}$$

and

$$V_1 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ x_2 & -y_2 & -z_2 & x_1 & -y_1 & -z_1 \\ y_2 & x_2 & 0 & y_1 & x_1 & 0 \\ z_2 & 0 & x_2 & z_1 & 0 & x_1 \end{pmatrix}, \tag{210}$$

$$V_2 = \mathrm{I}_6. \tag{211}$$

C.3 Two-Target Sum-of-Powers

For the two-target case, R_{pow} can be approximated as

$$R \approx \sum_{i=1}^{3} V_i q_i q_i^T V_i^T, \tag{212}$$

where

$$q_1^T = [u_{x_1} \quad u_{y_1} \quad u_{z_1} \quad u_{x_2} \quad u_{y_2} \quad u_{z_2}], \tag{213}$$

$$q_2^T = [u_{x_1}^2 \quad u_{y_1}^2 \quad u_{z_1}^2 \quad u_{x_2}^2 \quad u_{y_2}^2 \quad u_{z_2}^2 \quad], \tag{214}$$

$$q_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2\sigma^2 \\ 2(u_{x_1}u_{y_1} + u_{x_2}u_{y_2}) \\ 2(u_{x_1}u_{z_1} + u_{x_2}u_{z_2}) \end{pmatrix}, \tag{215}$$

and

$$V_1 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 2x_1 & -2y_1 & -2z_1 & 2x_2 & -2y_2 & -2z_2 \\ 2y_1 & 2x_1 & 0 & 2y_2 & 2x_2 & 0 \\ 2z_1 & 0 & 2x_1 & 2z_2 & 0 & 2x_2 \end{pmatrix}, \tag{216}$$

$$V_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{217}$$

$$V_3 = \mathrm{I}_6. \tag{218}$$

C.4 Jacobian Matrix

The Jacobian matrix, $\mathcal{J}$, for the three-dimensional, two-target sum-of-products SME is given by

$$\mathcal{J}_{prod} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ x_2 & -y_2 & -z_2 & x_1 & -y_1 & -z_1 \\ y_2 & x_2 & 0 & y_1 & x_1 & 0 \\ z_2 & 0 & x_2 & z_1 & 0 & x_1 \end{pmatrix}, \tag{219}$$

and the Jacobian matrix, $\mathcal{J}$, for the three-dimensional, two-target sum-of-powers SME is given by

$$\mathcal{J}_{pow} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 2x_1 & -2y_1 & -2z_1 & 2x_2 & -2y_2 & -2z_2 \\ 2y_1 & 2x_1 & 0 & 2y_2 & 2x_2 & 0 \\ 2z_1 & 0 & 2x_1 & 2z_2 & 0 & 2x_2 \end{pmatrix}. \tag{220}$$

APPENDIX D

SUPPLEMENT TO CHAPTER V: DERIVING TAYLOR'S EKF-BASED CRAMÉR-RAO LOWER BOUND

Taylor presents his EKF-based method for computing the CRLB in a brief paper [36]. He relies upon the identity

$$P_k^{-1} = [\Phi P_{k-1} \Phi^T + Q]^{-1} + H_k^T R_k^{-1} H_k, \tag{221}$$

and cites [12] as the source. While [12] does provide this identity and a description of how to find it, neither place works through the derivation. In this appendix, we work through the derivation in the hopes of gaining insight into this method of computing a CRLB. We begin with the following two lemmas.

Lemma 1.

$$P_k^{-1} = (P_k^-)^{-1} + H_k^T R_k^{-1} H_k. \tag{222}$$

Proof. We verify (222) using the identity

$$P_k P_k^{-1} = I, \tag{223}$$

where I is the identity matrix, and, from the EKF,

$$P_k = [I - K_k H_k] P_k^-, \tag{224}$$

where K_k is the Kalman gain defined by

$$K_k = P_k^- H_k^T (R_k + H_k P_k H_k^T)^{-1}. \tag{225}$$

For simplicity, we drop the k subscript on all terms in this proof. Substituting (224) and (222) into the left hand side of (223), we get

$$[(I - K_k H_k) P_k^-][(P_k^-)^{-1} + H_k^T R_k^{-1} H_k] =$$

$$
\begin{aligned}
&= (I - KH)P^-(P^-)^{-1} + (I - KH)P^- \\
&= (I - KH) + P^-H^TR^{-1}H - KHP^-H^TR^{-1}H \\
&= I - (K - P^-H^TR^{-1} + KHP^-H^TR^{-1})H \\
&= I - (K(I + HP^-H^TR^{-1}) - P^-H^TR^{-1})H \\
&= I - (K(R + HP^-H^T)R^{-1} - P^-H^TR^{-1})H \\
&= I - (P^-H^T(R + HP^-H^T)^{-1}(R + HP^-H^T)R^{-1} - P^-H^TR^{-1})H \qquad \text{Recall (225)} \\
&= I - (P^-H^TR^{-1} - P^-H^TR^{-1})H \\
&= I.
\end{aligned}
$$

□

Lemma 2.

$$K_k = P_k^-H_k^T[H_kP_k^-H_k^T + R_k]^{-1}. \tag{226}$$

Proof. Again, we drop the k subscript for simplification.

$$
\begin{aligned}
K &= IK \\
&= PP^{-1}K \\
&= P\left((P^-)^{-1} + H^TR^{-1}H\right)K \qquad \text{Lemma 1} \\
&= P\left((P^-)^{-1}K + H^TR^{-1}HK\right) \\
&= P\left((P^-)^{-1}P^-H^T(R + HP^-H^T)^{-1} + H^TR^{-1}HK\right) \qquad \text{Recall (225)} \\
&= P\left[H^T(R + HP^-H^T)^{-1} + H^TR^{-1}HP^-H^T(R + HP^-H^T)^{-1}\right] \qquad \text{Recall (225)} \\
&= P\left[H^T + H^TR^{-1}HP^-H^T\right]\left[R + HP^-H^T\right]^{-1} \\
&= PH^T\left[I + R^{-1}HP^-H^T\right]\left[R + HP^-H^T\right]^{-1} \\
&= PH^TR^{-1}R\left[I + R^{-1}HP^-H^T\right]\left[R + HP^-H^T\right]^{-1} \\
&= PH^TR^{-1}\left[R + HP^-H^T\right]\left[R + HP^-H^T\right]^{-1} \\
&= PH^TR^{-1}
\end{aligned}
$$

□

We are now ready to proceed deriving (221). From the EKF equations,

$$P_k = [I - K_k H_k] P_k^-, \tag{227}$$

$$P_k^- = \Phi_{k-1} P_{k-1} \Phi_{k-1}^T + Q_{k-1}. \tag{228}$$

Substituting Lemma 2 yields

$$\begin{aligned}
P_k &= [I - K_k H_k] P_k^-, \\
P_k &= [I - P_k H_k^T R_k^{-1} H_k] P_k^-, \\
P_k &= P_k [P_k^{-1} - H_k^T R_k^{-1} H_k] P_k^-, \\
I &= [P_k^{-1} - H_k^T R_k^{-1} H_k] P_k^-, \\
(P_k^-)^{-1} &= P_k^{-1} - H_k^T R_k^{-1} H_k, \\
P_k^{-1} &= (P_k^-)^{-1} + H_k^T R_k^{-1} H_k, \\
P_k^{-1} &= (\Phi_{k-1} P_{k-1} \Phi_{k-1}^T + Q_{k-1})^{-1} + H_k^T R_k^{-1} H_k, && \text{Substitute (228)} \\
P_k^{-1} &= (\Phi_{k-1} P_{k-1} \Phi_{k-1}^T)^{-1} + H_k^T R_k^{-1} H_k. && \text{Assume } Q_{k-1} = 0
\end{aligned}$$

REFERENCES

[1] Balzarotti, G. and Marsiglia, G., "Quaternions in line-of-sight control," *Optical Engineering*, vol. 44, no. 10, p. 103003, 2005.

[2] Bar-Shalom, Y., Li, X.-R., and Kirubarajan, T., *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, 2001.

[3] Bar-Shalom, Y. and Li, X., *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.

[4] Bellaire, R., *Nonlinear Estimation with Applications to Target Tracking*. PhD thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 1996.

[5] Blackman, S. and Popoli, R., *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[6] Borden, R. S., *A Course in Advanced Calculus*. Reading, NY: Dover Publications, 1998.

[7] Bruno, M. G. and Pavlov, A., "Improved particle filters for ballistic target tracking," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 2, pp. 705–8, May 2004.

[8] Daum, F., "Exact finite-dimensional nonlinear filters," vol. 31, pp. 616–622, 1986.

[9] Daum, F., "Bounds on the performace of multiple target tracking," *IEEE Trans. on Automatic Control*, vol. 35, April 1990.

[10] Daum, F., "A Cramér-Rao bound for multiple target tracking," in *Signal and Data Processing of Small Targets* (Drummond, O., ed.), vol. SPIE Proc. 1481, (Orlando, FL), pp. 341–344, Apr. 1991.

[11] Doucet, A., de Freitas, N., and Gordon, N., *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[12] Gelb, A., ed., *Applied Optimal Estimation*. Cambridge, MA: M.I.T. Press, 1974.

[13] Hue, C., Le Cadre, J.-P., and Pèrez, P., "Performance analysis of two sequential Monte Carlo methods and posterior Cramer-Rao bounds for multi-target tracking," in *Proceedings of the Fifth International Conference on Information Fusion*, (Annapolis, Maryland), pp. 464 – 473, 2002.

[14] Ito, K. and Xiong, K., "Gaussian filters for nonlinear filtering problems," vol. 45, pp. 910–927, May 2000.

[15] Julier, S. J., "The scaled unscented transformation," in *Proceedings of the 2002 American Control Conference*, vol. 6, (Anchorage, AK), pp. 4555 – 4559, May 2002.

[16] Julier, S. J. and Uhlmann, J. K., "New extension of the Kalman filter to nonlinear systems," in *Proceedings of the SPIE*, (Orlando, FL), pp. 182–193, SPIE, Apr. 1997.

[17] Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F., "A new method for the nonlinear transformation of means and covariances in filters and estimators," vol. 45, pp. 477–482, Mar. 2000.

[18] Kamen, E., "Multiple target tracking based on symmetric measurement equations," vol. 37, pp. 371–374, April 1992.

[19] Kamen, E. and Sastry, C., "Multiple target tracking using products of position measurements," vol. 29, no. 2, pp. 476–493, 1993.

[20] Kamen, E. and Su, J., *Introduction to Optimal Estimation*. Springer-Verlag, 1999.

[21] Karlsson, R., Gustafsson, F., and Karlsson, T., "Particle filtering and Cramér-Rao lower bound for underwater navigation," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). IEEE International Conference on*, vol. 6, pp. 65–8, Apr. 2003.

[22] Kastella, K., "Mean-field theory and multitarget tracking," in *Signal and Data Processing of Small Targets* (Drummond, O., ed.), vol. SPIE Proc. 2235, pp. 388–393, July 1994.

[23] Kastella, K., "Event averaged maximum likelihood estimation and mean-field theory in multitarget tracking," *IEEE Trans. on Automatic Control*, vol. 40, no. 6, pp. 1070–1074, 1995.

[24] Lee, Y., *The SME Filter Approach to Multiple Target Tracking with False and Missing Measurements*. PhD thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 1994.

[25] Lefebvre, T., Bruyninckx, H., and Schuller, J. D., "Comment on 'A new method for the nonlinear transformation of means and covariances in filters and estimators' [and authors' reply]," vol. 47, pp. 1406– 1409, Aug. 2002.

[26] Leven, W. F. and Lanterman, A. D., "Multiple target tracking with symmetric measurement equations using unscented Kalman and particle filters," in *Proceedings of the* 36^{th} *Southeastern Symposium on System Theory*, pp. 195–199, Mar. 2004.

[27] Leven, W. F. and Lanterman, A. D., "Unscented Kalman filters for multiple target tracking with symmetric measurement equations," submitted to IEEE Trans. on Automatic Control Jan 2005.

[28] Mahler, R., "Objective functions for Bayesian control-theoretic sensor management, I: Multitarget first-moment approximation," in *Proc. of IEEE Aerospace Conference*, pp. 1905–1924, Mar. 2003.

[29] Mahler, R. P. S., "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, pp. 1152–1178, Oct. 2003.

[30] Mauroy, G. P., *Multiple Target Tracking using Neural Networks and Set Estimation*. PhD thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 1997.

[31] Muder, D. and O'Neil, S., "The multi-dimensional SME filter for multitarget tracking," in *Signal and Data Processing of Small Targets* (Drummond, O., ed.), vol. SPIE Proc. 1954, pp. 587–599, 1993.

[32] Nørgaard, M., Poulsen, N. K., and Ravn, O., "Advances in derivative-free state estimation for nonlinear systems," technical report imm-rep-1998-15 (revised edition), Technical University of Denmark, Denmark, Apr. 2000.

[33] Orton, M. and Fitzgerald, W., "A Bayesian approach to tracking multiple targets using sensor arrays and particle filters," vol. 50, pp. 216–223, Feb. 2002.

[34] Papoulis, A. and Pillai, S. U., *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, Inc., fourth ed., 2002.

[35] Sastry, C. and Kamen, E., "SME filter approach to multiple target tracking with radar measurements," *IEE Proc. F: Radar, Sonar, and Navigation*, vol. 140, pp. 251–260, August 1993.

[36] Taylor, J. H., "The Cramér-Rao estimation error lower bound computation for deterministic nonlinear systems," vol. 24, pp. 343–344, Apr. 1979.

[37] Taylor Jr., R. M., Flanagan, B. P., and Uber, J. A., "Computing the recursive posterior Cramér-Rao bound for a radar tracking system," working paper, The MITRE Co., McLean, VA, 2005.

[38] Tichavský, P., Muravchik, C., and Nehorai, A., "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. on Signal Proc.*, vol. 40, pp. 1386–1396, May 1998.

[39] Tobias, M. and Lanterman, A. D., "Probability hypothesis density-based multitarget tracking with bistatic range and Doppler observations," *IEE Proc. Radar, Sonar and Navigation*, vol. 152, pp. 195–205, June 2005.

[40] Van Trees, H. L., *Detection, Estimation and Modulation Theory, Part I*. New York: John Wiley and Sons, 1968.

[41] Wan, E. and van der Merwe, R., *Kalman Filtering and Neural Networks*, ch. 7. Wiley, 2001.

[42] Wan, E. and van der Merwe, R., *Recursive Bayesian Estimation Library*. Beaverton, OR: OGI School of Science and Engineering, 2003.

[43] Wan, E. A. and van der Merwe, R., "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000 (AS-SPCC)*, (Lake Louise, Alberta, Canada), pp. 153–158, The IEEE, Oct. 2000.

[44] Wan, E. A. and van der Merwe, R., "The square-root unscented Kalman filter for nonlinear estimation," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, (Salt Lake City, Utah), pp. 3461–4, May 2001.

[45] Wikipedia, "Quaternion — wikipedia, the free encyclopedia," 2006. [Online; accessed 8-January-2006].

VITA

William Leven received his B.S. Degree in Electrical Engineering and Computer Science from the University of California, Berkeley in 2002 and his M.S. Degree in Electrical and Computer Engineering from the Georgia Institute of Technology in 2004. He received his Ph.D. Degree in Electrical and Computer Engineering from the Georgia Institute of Technology in 2006. His graduate research focused on multiple target tracking and the computation of bounds for the multiple target tracking problem. Outside of his graduate studies, he has spent several internships at Texas Instruments Inc. developing signal processing software for digital subscriber line (DSL) modems and designing algorithms for next-generation high-speed serial communication links.